

Index

Cocos Creator v2.0.0

Welcome to Cocos Creator JavaScript engine API reference. You can search anything in top left search filed.

All enums and classes are under `cc` module if not specified otherwise.

Modules

- [GLOBAL-MACROS](#)
- [_decorator](#)
- [anysdk](#)
- [cc](#)
- [dragonBones](#)
- [js](#)
- [sp](#)
- [sp.spine](#)

Classes

- [Action](#) from [cc](#)
- [ActionInstant](#) from [cc](#)
- [ActionInterval](#) from [cc](#)
- [ActionManager](#) from [cc](#)
- [AffineTransform](#) from [cc](#)
- [AgentManager](#) from [anysdk](#)
- [Animation](#) from [cc](#)
- [Animation.EventType](#) from [cc](#)
- [AnimationClip](#) from [cc](#)
- [AnimationState](#) from [cc](#)
- [ArmatureDisplay](#) from [dragonBones](#)
- [array](#) from [js](#)
- [Asset](#) from [cc](#)
- [AudioClip](#) from [cc](#)
- [audioEngine](#) from [cc](#)
- [AudioSource](#) from [cc](#)
- [BitmapFont](#) from [cc](#)
- [BlockInputEvents](#) from [cc](#)
- [BoxCollider](#) from [cc](#)
- [Button](#) from [cc](#)
- [CallbacksInvoker](#) from [cc](#)
- [Camera](#) from [cc](#)
- [Canvas](#) from [cc](#)
- [CircleCollider](#) from [cc](#)

- `CoffeeScript` from `cc`
- `Collider` from `cc`
- `Collider.Box` from `cc`
- `Collider.Circle` from `cc`
- `Collider.Polygon` from `cc`
- `CollisionManager` from `cc`
- `Color` from `CC`
- `Component` from `cc`
- `Component.EventHandler` from `cc`
- `constructor` from `CC`
- `ContainerStrategy` from `cc`
- `ContentStrategy` from `cc`
- `debug` from `cc`
- `Details` from `cc`
- `Director` from `cc`
- `DistanceJoint` from `cc`
- `DragonBonesAsset` from `dragonBones`
- `DragonBonesAtlasAsset` from `dragonBones`
- `EditBox` from `cc`
- `EqualToFrame` from `cc`
- `EqualToWindow` from `cc`
- `Event` from `CC`
- `Event.EventAcceleration` from `cc`
- `Event.EventCustom` from `cc`
- `Event.EventKeyboard` from `cc`
- `Event.EventMouse` from `cc`
- `Event.EventTouch` from `cc`
- `EventTarget` from `cc`
- `FiniteTimeAction` from `cc`
- `Font` from `cc`
- `game` from `cc`
- `Graphics` from `cc`
- `Intersection` from `cc`
- `Joint` from `cc`
- `JsonAsset` from `cc`
- `Label` from `cc`
- `LabelAtlas` from `cc`
- `LabelOutline` from `cc`
- `Layout` from `cc`
- `loader` from `cc`
- `LoadingItems` from `cc`
- `macro` from `cc`
- `Manifold` from `cc`
- `ManifoldPoint` from `cc`
- `Mask` from `cc`
- `misc` from `cc`

- `MotionStreak` from [cc](#)
- `MotorJoint` from [cc](#)
- `MouseJoint` from [cc](#)
- `Node` from [cc](#)
- `Node.EventType` from [cc](#)
- `NodePool` from [cc](#)
- `OriginalContainer` from [cc](#)
- `PageView` from [cc](#)
- `PageViewIndicator` from [cc](#)
- `ParticleAsset` from [cc](#)
- `ParticleSystem` from [cc](#)
- `path` from [cc](#)
- `PhysicsBoxCollider` from [cc](#)
- `PhysicsChainCollider` from [cc](#)
- `PhysicsCircleCollider` from [cc](#)
- `PhysicsCollider` from [cc](#)
- `PhysicsContact` from [cc](#)
- `PhysicsImpulse` from [cc](#)
- `PhysicsManager` from [cc](#)
- `PhysicsPolygonCollider` from [cc](#)
- `PhysicsRayCastResult` from [cc](#)
- `Pipeline` from [cc](#)
- `Pipeline.Downloader` from [cc](#)
- `Pipeline.Loader` from [cc](#)
- `Playable` from [cc](#)
- `PluginParam` from [anysdk](#)
- `PluginProtocol` from [anysdk](#)
- `PolygonCollider` from [cc](#)
- `Pool` from [js](#)
- `Prefab` from [cc](#)
- `PrismaticJoint` from [cc](#)
- `PrivateNode` from [cc](#)
- `ProgressBar` from [cc](#)
- `ProportionalToFrame` from [cc](#)
- `ProportionalToWindow` from [cc](#)
- `ProtocolAds` from [anysdk](#)
- `ProtocolAdTracking` from [anysdk](#)
- `ProtocolAnalytics` from [anysdk](#)
- `ProtocolCrash` from [anysdk](#)
- `ProtocolCustom` from [anysdk](#)
- `ProtocolIAP` from [anysdk](#)
- `ProtocolPush` from [anysdk](#)
- `ProtocolREC` from [anysdk](#)
- `ProtocolShare` from [anysdk](#)
- `ProtocolSocial` from [anysdk](#)
- `ProtocolUser` from [anysdk](#)

- `Quat` from `cc`
- `RawAsset` from `cc`
- `Rect` from `cc`
- `RenderComponent` from `cc`
- `renderer` from `cc`
- `RenderTexture` from `cc`
- `ResolutionPolicy` from `cc`
- `RevoluteJoint` from `cc`
- `RichText` from `cc`
- `RigidBody` from `cc`
- `RopeJoint` from `cc`
- `Scene` from `cc`
- `SceneAsset` from `cc`
- `Scheduler` from `cc`
- `screen` from `cc`
- `Scrollbar` from `cc`
- `ScrollView` from `cc`
- `Size` from `cc`
- `Skeleton` from `sp`
- `SkeletonData` from `sp`
- `Slider` from `cc`
- `Sprite` from `cc`
- `SpriteAtlas` from `cc`
- `SpriteFrame` from `cc`
- `sys` from `cc`
- `SystemEvent` from `cc`
- `SystemEvent.EventType` from `cc`
- `TextAsset` from `cc`
- `Texture2D` from `cc`
- `TiledLayer` from `cc`
- `TiledMap` from `cc`
- `TiledMapAsset` from `cc`
- `TiledObjectGroup` from `cc`
- `TiledTile` from `cc`
- `Toggle` from `cc`
- `ToggleContainer` from `cc`
- `ToggleGroup` from `cc`
- `Touch` from `cc`
- `TTFFont` from `cc`
- `TypeScript` from `cc`
- `url` from `cc`
- `ValueType` from `cc`
- `Vec2` from `cc`
- `Vec3` from `cc`
- `VideoPlayer` from `cc`
- `View` from `cc`

- `ViewGroup` from [cc](#)
- `visibleRect` from [cc](#)
- `WebView` from [cc](#)
- `WeldJoint` from [cc](#)
- `WheelJoint` from [cc](#)
- `Widget` from [cc](#)
- `WorldManifold` from [cc](#)
- `WXSubContextView` from [cc](#)

Enums

- `AccountGender` from [anysdk](#)
- `AccountOperate` from [anysdk](#)
- `AccountType` from [anysdk](#)
- `AdsPos` from [anysdk](#)
- `AdsResultCode` from [anysdk](#)
- `AdsType` from [anysdk](#)
- `AnimationEventType` from [sp](#)
- `audioEngine.AudioState` from [cc](#)
- `Button.Transition` from [cc](#)
- `Camera.ClearFlags` from [cc](#)
- `CustomresultCode` from [anysdk](#)
- `debug.DebugMode` from [cc](#)
- `DrawBits` from [cc](#)
- `EditBox.InputFlag` from [cc](#)
- `EditBox.InputMode` from [cc](#)
- `EditBox.KeyboardReturnType` from [cc](#)
- `Graphics.LineCap` from [cc](#)
- `Graphics.LineJoin` from [cc](#)
- `Label.HorizontalAlign` from [cc](#)
- `Label.Overflow` from [cc](#)
- `Label.Type` from [cc](#)
- `Label.VerticalAlign` from [cc](#)
- `Layout.AxisDirection` from [cc](#)
- `Layout.HorizontalDirection` from [cc](#)
- `Layout.ResizeMode` from [cc](#)
- `Layout.Type` from [cc](#)
- `Layout.VerticalDirection` from [cc](#)
- `LoadingItems.ItemState` from [cc](#)
- `macro.BlendFactor` from [cc](#)
- `macro.ImageFormat` from [cc](#)
- `macro.KEY` from [cc](#)
- `macro.TextAlignment` from [cc](#)
- `Mask.Type` from [cc](#)
- `NetworkType` from [cc](#)
- `PageView.Direction` from [cc](#)

- `PageView.EventType` from [cc](#)
- `PageView.SizeMode` from [cc](#)
- `PageViewIndicator.Direction` from [cc](#)
- `ParticleSystem.EmitterMode` from [cc](#)
- `ParticleSystem.PositionType` from [cc](#)
- `PayResultCode` from [anysdk](#)
- `Prefab.OptimizationPolicy` from [cc](#)
- `ProgressBar.Mode` from [cc](#)
- `PushActionResultCode` from [anysdk](#)
- `RayCastType` from [cc](#)
- `RECResultCode` from [anysdk](#)
- `RigidBodyType` from [cc](#)
- `Scrollbar.Direction` from [cc](#)
- `ScrollView.EventType` from [cc](#)
- `ShareResultCode` from [anysdk](#)
- `Slider.Direction` from [cc](#)
- `SocialRetCode` from [anysdk](#)
- `Sprite.FillType` from [cc](#)
- `Sprite.SizeMode` from [cc](#)
- `Sprite.Type` from [cc](#)
- `TaskType` from [anysdk](#)
- `Texture2D.Filter` from [cc](#)
- `Texture2D.PixelFormat` from [cc](#)
- `Texture2D.WrapMode` from [cc](#)
- `TiledMap.Orientation` from [cc](#)
- `ToolBarPlace` from [anysdk](#)
- `UserActionResultCode` from [anysdk](#)
- `VerticalTextAlignment` from [cc](#)
- `VideoPlayer.EventType` from [cc](#)
- `VideoPlayer.ResourceType` from [cc](#)
- `WebView.EventType` from [cc](#)
- `Widget.AlignMode` from [cc](#)
- `WrapMode` from [cc](#)

Editor API

- [AssetDB](#)

Main Process

- [Editor](#)
- [Editor \(Console Module\)](#)
- [Editor.App](#)
- [Editor.Debugger](#)
- [Editor.DevTools](#)

- [Editor.Dialog](#)
- [Editor.Ipc](#)
- [Editor.MainMenu](#)
- [Editor.Menu](#)
- [Editor.Package](#)
- [Editor.Panel](#)
- [Editor.Profile](#)
- [Editor.Protocol](#)
- [Editor.Window](#)
- [Editor.Worker](#)

Renderer Process (Web Page)

- [Editor](#)
- [Editor \(Console Module\)](#)
- [Editor.Audio](#)
- [Editor.Dialog](#)
- [Editor.Ipc](#)
- [Editor.MainMenu](#)
- [Editor.Menu](#)
- [Editor.Package](#)
- [Editor.Panel](#)
- [Editor.Protocol](#)
- [Editor.Window](#)
- [Editor.UI \(DOM Utils Module\)](#)
- [Editor.UI \(Element Utils Module\)](#)
- [Editor.UI \(Focus Module\)](#)
- [Editor.UI \(Resources Module\)](#)
- [Editor.UI.Settings](#)
- [Editor.UI.DockUtils](#)
- [Editor.UI.DragDrop](#)

Modules for Both Processes

- [Editor](#)
- [Editor.Easing](#)
- [Editor.IpcListener](#)
- [Editor.JS](#)
- [Editor.Math](#)
- [Editor.Selection](#)
- [Editor.Undo](#)
- [Editor.Utils](#)
- [Editor.i18n](#)

GLOBAL-MACROS Module

Here are some of the macro used to determine the execution environment, these macros are global variables, can be accessed directly.

When the project is built, these macros will be preprocessed and discard unreachable code based on the built platform, for example:

```
if (CC_DEBUG) {  
    cc.log('debug');  
}  
else {  
    cc.log('release');  
}
```

After build will become:

```
cc.log('release');
```

To determine whether the script is running on the specified platform, you can use the following expression:

```
{  
    "editor": CC_EDITOR,  
    "editor or preview": CC_DEV,  
    "editor or preview or build in debug mode": CC_DEBUG,  
    "web preview": CC_PREVIEW && !CC_JSB,  
    "simulator preview": CC_PREVIEW && CC_JSB,  
    "build in debug mode": CC_BUILD && CC_DEBUG,  
    "build in release mode": CC_BUILD && !CC_DEBUG,  
}
```

Index

Properties

- [CC_EDITOR](#) Boolean Running in the editor.
- [CC_PREVIEW](#) Boolean Preview in browser or simulator.
- [CC_DEV](#) Boolean Running in the editor or preview.
- [CC_DEBUG](#) Boolean Running in the editor or preview, or build in debug mode.
- [CC_BUILD](#) Boolean Running in published project.
- [CC_JSB](#) Boolean Running in native platform (mobile app, desktop app, or simulator).
- [CC_TEST](#) Boolean Running in the engine's unit test.
- [CC_WECHATGAME](#) Boolean Running in the Wechat's mini game.
- [CC_QQPLAY](#) Boolean Running in the bricks.
- [CC_RUNTIME](#) Boolean Running in runtime environments.

Details

Properties

CC_EDITOR

Running in the editor.

meta	description
Type	Boolean
Defined in	predefine.js:90

CC_PREVIEW

Preview in browser or simulator.

meta	description
Type	Boolean
Defined in	predefine.js:93

CC_DEV

Running in the editor or preview.

meta	description
Type	Boolean
Defined in	predefine.js:96

CC_DEBUG

Running in the editor or preview, or build in debug mode.

meta	description
Type	Boolean
Defined in	predefine.js:99

CC_BUILD

Running in published project.

meta	description
Type	Boolean
Defined in	predefine.js:102

CC_JSB

Running in native platform (mobile app, desktop app, or simulator).

meta	description
Type	Boolean
Defined in	predefine.js:105

CC_TEST

Running in the engine's unit test.

meta	description
Type	Boolean
Defined in	predefine.js:108

CC_WECHATGAME

Running in the Wechat's mini game.

meta	description
Type	Boolean

meta	description
Defined in	predefine.js:111

CC_QQPLAY

Running in the bricks.

meta	description
Type	Boolean
Defined in	predefine.js:114

CC_RUNTIME

Running in runtime environments.

meta	description
Type	Boolean
Defined in	predefine.js:117

_decorator Module

Some JavaScript decorators which can be accessed with "cc._decorator".

Index

Methods

- `ccclass` Declare the standard [ES6 Class](#)...
- `property` Declare property for [CCClass](#).
- `executeInEditMode` Makes a CCClass that inherit from component execute in edit mode.
...
- `requireComponent` Automatically add required component as a dependency for the CCClass that inherit from component.

- `menu` The menu path to register a component to the editors "Component" menu.
- `executionOrder` The execution order of lifecycle methods for Component.
- `disallowMultiple` Prevents Component of the same type (or subtype) to be added more than once to a Node.
- `playOnFocus` If specified, the editor's scene view will keep updating this node in 60 fps when it is selected, otherwise, it will update only if necessary.
- `...`
- `inspector` Specifying the url of the custom html to draw the component in **Properties**.
- `icon` Specifying the url of the icon to display in the editor.
- `help` The custom documentation URL.
- `mixins` The old mixins implemented in cc.Class(ES5) behaves exact the same as multiple inheritance.

Details

Methods

ccclass

Declare the standard [ES6 Class](#) as CCClass, please see [Class](#) for details.

meta	description
Defined in	cocos2d/core/platform/CCClassDecorator.js:243

Parameters

- `name` [String](#) The class name used for serialization.

Examples

```
const {ccclass} = cc._decorator;

// define a CCClass, omit the name
@ccclass
class NewScript extends cc.Component {
    // ...
}

// define a CCClass with a name
@ccclass('LoginData')
class LoginData {
    // ...
}
```

property

Declare property for [CCClass](#).

meta	description
Defined in	cocos2d/core/platform/CCClassDecorator.js:315

Parameters

- options [Object](#) an object with some property attributes
- type [Any](#)
- visible [Boolean](#) | [Function](#)
- displayName [String](#)
- tooltip [String](#)
- multiline [Boolean](#)
- readonly [Boolean](#)
- min [Number](#)
- max [Number](#)
- step [Number](#)
- range [Number\[\]](#)
- slide [Boolean](#)
- serializable [Boolean](#)
- editorOnly [Boolean](#)
- override [Boolean](#)
- animatable [Boolean](#)
- formerlySerializedAs [String](#)

Examples

```
const {ccclass, property} = cc._decorator;

@ccclass
class NewScript extends cc.Component {
    @property({
        type: cc.Node
    })
    targetNode1 = null;

    @property(cc.Node)
    targetNode2 = null;

    @property(cc.Button)
    targetButton = null;

    @property
    _width = 100;

    @property
```

```
get width () {
    return this._width;
}

@property
set width (value) {
    this._width = value;
}

@property
offset = new cc.Vec2(100, 100);

@property(cc.Vec2)
offsets = [];

@property(cc.SpriteFrame)
frame = null;
}

// above is equivalent to (上面的代码相当于):

var NewScript = cc.Class({
    properties: {
        targetNode1: {
            default: null,
            type: cc.Node
        },

        targetNode2: {
            default: null,
            type: cc.Node
        },

        targetButton: {
            default: null,
            type: cc.Button
        },

        _width: 100,

        width: {
            get () {
                return this._width;
            },
            set (value) {
                this._width = value;
            }
        },
        offset: new cc.Vec2(100, 100)

        offsets: {
            default: [],
            type: cc.Vec2
        }
    }
})
```

```

        }

        frame: {
            default: null,
            type: cc.SpriteFrame
        },
    },
});
```

executeInEditMode

Makes a CCClass that inherit from component execute in edit mode.

By default, all components are only executed in play mode, which means they will not have their callback functions executed while the Editor is in edit mode.

meta	description
Defined in	cocos2d/core/platform/CCClassDecorator.js:461

Examples

```

const {ccclass, executeInEditMode} = cc._decorator;

@ccclass
@executeInEditMode
class NewScript extends cc.Component {
    // ...
}
```

requireComponent

Automatically add required component as a dependency for the CCClass that inherit from component.

meta	description
Defined in	cocos2d/core/platform/CCClassDecorator.js:485

Parameters

- `requiredComponent Component`

Examples

```

const {ccclass, requireComponent} = cc._decorator;
```

```
@ccclass
@requireComponent(cc.Sprite)
class SpriteCtrl extends cc.Component {
    // ...
}
```

menu

The menu path to register a component to the editors "Component" menu. Eg. "Rendering/CameraCtrl".

meta	description
Defined in	cocos2d/core/platform/CCClassDecorator.js:506

Parameters

- **path [String](#)** The path is the menu represented like a pathname.
For example the menu could be "Rendering/CameraCtrl".

Examples

```
const {ccclass, menu} = cc._decorator;

@ccclass
@menu("Rendering/CameraCtrl")
class NewScript extends cc.Component {
    // ...
}
```

executionOrder

The execution order of lifecycle methods for Component. Those less than 0 will execute before while those greater than 0 will execute after. The order will only affect onLoad, onEnable, start, update and lateUpdate while onDisable and onDestroy will not be affected.

meta	description
Defined in	cocos2d/core/platform/CCClassDecorator.js:528

Parameters

- **order [Number](#)** The execution order of lifecycle methods for Component. Those less than 0 will execute before while those greater than 0 will execute after.

Examples

```
const {ccclass, executionOrder} = cc._decorator;

@ccclass
@executionOrder(1)
class CameraCtrl extends cc.Component {
    // ...
}
```

DisallowMultiple

Prevents Component of the same type (or subtype) to be added more than once to a Node.

meta	description
Defined in	cocos2d/core/platform/CCClassDecorator.js:551

Examples

```
const {ccclass, disallowMultiple} = cc._decorator;

@ccclass
@disallowMultiple
class CameraCtrl extends cc.Component {
    // ...
}
```

playOnFocus

If specified, the editor's scene view will keep updating this node in 60 fps when it is selected, otherwise, it will update only if necessary.

This property is only available if executeInEditMode is true.

meta	description
Defined in	cocos2d/core/platform/CCClassDecorator.js:572

Examples

```
const {ccclass, playOnFocus, executeInEditMode} = cc._decorator;

@ccclass
@executeInEditMode
@playOnFocus
class CameraCtrl extends cc.Component {
```

```
// ...  
}
```

inspector

Specifying the url of the custom html to draw the component in **Properties**.

meta	description
Defined in	cocos2d/core/platform/CCClassDecorator.js:595

Parameters

- `url` [String](#)

Examples

```
const {ccclass, inspector} = cc._decorator;  
  
@ccclass  
@inspector("packages://inspector/inspectors/comps/camera-ctrl.js")  
class NewScript extends cc.Component {  
    // ...  
}
```

icon

Specifying the url of the icon to display in the editor.

meta	description
Defined in	cocos2d/core/platform/CCClassDecorator.js:616

Parameters

- `url` [String](#)

Examples

```
const {ccclass, icon} = cc._decorator;  
  
@ccclass  
@icon("xxxx.png")  
class NewScript extends cc.Component {  
    // ...  
}
```

help

The custom documentation URL.

meta	description
Defined in	cocos2d/core/platform/CCClassDecorator.js:638

Parameters

- `url` [String](#)

Examples

```
const {ccclass, help} = cc._decorator;

@ccclass
@help("app://docs/html/components/spine.html")
class NewScript extends cc.Component {
    // ...
}
```

mixins

NOTE:

The old mixins implemented in cc.Class(ES5) behaves exact the same as multiple inheritance. But since ES6, class constructor can't be function-called and class methods become non-enumerable, so we can not mix in ES6 Classes.

See:

<https://esdiscuss.org/topic/traits-are-now-impossible-in-es6-until-es7-since-rev32>

One possible solution (but IDE unfriendly):

<http://justinfagnani.com/2015/12/21/real-mixins-with-javascript-classes>

NOTE:

You must manually call mixins constructor, this is different from cc.Class(ES5).

meta	description
Defined in	cocos2d/core/platform/CCClassDecorator.js:661

Parameters

- `ctor` [Function](#) constructors to mix, only support ES5 constructors or classes defined by using `cc.Class`,

- not support ES6 Classes.

Examples

```
const {ccclass, mixins} = cc._decorator;

class Animal { ... }

const Fly = cc.Class({
    constructor () { ... }
});

@ccclass
@mixin(cc.EventTarget, Fly)
class Bird extends Animal {
    constructor () {
        super();

        // You must manually call mixins constructor, this is different from
        cc.Class(ES5)
        cc.EventTarget.call(this);
        Fly.call(this);
    }
    // ...
}
```

anysdk Module

AnySDK is a third party solution that offers game developers SDK integration without making changes to the SDK's features or parameters. It can do all of this while remaining invisible to your end user. Our goal is to handle all the tedious SDK integration work for you so that you can use your time to focus on the game itself. No matter if it's the channel SDK, user system, payment system, ad system, statistics system, sharing system or any other type of SDK: we'll take care of it for you.

Classes

- [AgentManager](#)
- [PluginParam](#)
- [PluginProtocol](#)
- [ProtocolAds](#)
- [ProtocolAdTracking](#)
- [ProtocolAnalytics](#)
- [ProtocolCrash](#)
- [ProtocolCustom](#)
- [ProtocolIAP](#)
- [ProtocolPush](#)
- [ProtocolREC](#)
- [ProtocolShare](#)

- [ProtocolSocial](#)
- [ProtocolUser](#)

Enums

- [AccountGender](#)
- [AccountOperate](#)
- [AccountType](#)
- [AdsPos](#)
- [AdsresultCode](#)
- [AdsType](#)
- [AnimationEventType](#)
- [audioEngine.AudioState](#)
- [Button.Transition](#)
- [Camera.ClearFlags](#)
- [CustomresultCode](#)
- [debug.DebugMode](#)
- [DrawBits](#)
- [EditBox.InputFlag](#)
- [EditBox.InputMode](#)
- [EditBox.KeyboardReturnType](#)
- [Graphics.LineCap](#)
- [Graphics.LineJoin](#)
- [Label.HorizontalAlign](#)
- [Label.Overflow](#)
- [Label.Type](#)
- [Label.VerticalAlign](#)
- [Layout.AxisDirection](#)
- [Layout.HorizontalDirection](#)
- [Layout.ResizeMode](#)
- [Layout.Type](#)
- [Layout.VerticalDirection](#)
- [LoadingItems.ItemState](#)
- [macro.BlendFactor](#)
- [macro.ImageFormat](#)
- [macro.KEY](#)
- [macro.TextAlignment](#)
- [Mask.Type](#)
- [NetworkType](#)
- [PageView.Direction](#)
- [PageView.EventType](#)
- [PageView.SizeMode](#)
- [PageViewIndicator.Direction](#)
- [ParticleSystem.EmitterMode](#)
- [ParticleSystem.PositionType](#)
- [PayresultCode](#)
- [Prefab.OptimizationPolicy](#)
- [ProgressBar.Mode](#)
- [PushActionResultCode](#)
- [RayCastType](#)

- [RECResultCode](#)
- [RigidBodyType](#)
- [Scrollbar.Direction](#)
- [ScrollView.EventType](#)
- [ShareResultCode](#)
- [Slider.Direction](#)
- [SocialRetCode](#)
- [Sprite.FillType](#)
- [Sprite.SizeMode](#)
- [Sprite.Type](#)
- [TaskType](#)
- [Texture2D.Filter](#)
- [Texture2D.PixelFormat](#)
- [Texture2D.WrapMode](#)
- [TiledMap.Orientation](#)
- [ToolBarPlace](#)
- [UserActionResultCode](#)
- [VerticalTextAlignment](#)
- [VideoPlayer.EventType](#)
- [VideoPlayer.ResourceType](#)
- [WebView.EventType](#)
- [Widget.AlignMode](#)
- [WrapMode](#)

Index

Properties

- `agentManager` `anysdk.AgentManager` agent manager of plugin

Details

Properties

`agentManager`

agent manager of plugin

meta	description
Type	<code>anysdk.AgentManager</code>
Defined in	extensions/anysdk/jsb_anysdk.js:40

cc Module

The main namespace of Cocos2d-JS, all engine core classes, functions, properties and constants are defined in this namespace.

Classes

- [BaseNode](#)
- [CallbacksHandler](#)
- [JavaScript](#)
- [Script](#)
- [Action](#)
- [ActionInstant](#)
- [ActionInterval](#)
- [ActionManager](#)
- [AffineTransform](#)
- [Animation](#)
- [Animation.EventType](#)
- [AnimationClip](#)
- [AnimationState](#)
- [Asset](#)
- [AudioClip](#)
- [audioEngine](#)
- [AudioSource](#)
- [BitmapFont](#)
- [BlockInputEvents](#)
- [BoxCollider](#)
- [Button](#)
- [CallbacksInvoker](#)
- [Camera](#)
- [Canvas](#)
- [CircleCollider](#)
- [CoffeeScript](#)
- [Collider](#)
- [Collider.Box](#)
- [Collider.Circle](#)
- [Collider.Polygon](#)
- [CollisionManager](#)
- [Color](#)
- [Component](#)
- [Component.EventHandler](#)
- [constructor](#)
- [ContainerStrategy](#)
- [ContentStrategy](#)
- [debug](#)
- [Details](#)

- [Director](#)
- [DistanceJoint](#)
- [EditBox](#)
- [EqualToFrame](#)
- [EqualToWindow](#)
- [Event](#)
- [Event.EventAcceleration](#)
- [Event.EventCustom](#)
- [Event.EventKeyboard](#)
- [Event.EventMouse](#)
- [Event.EventTouch](#)
- [EventTarget](#)
- [FiniteTimeAction](#)
- [Font](#)
- [game](#)
- [Graphics](#)
- [Intersection](#)
- [Joint](#)
- [JsonAsset](#)
- [Label](#)
- [LabelAtlas](#)
- [LabelOutline](#)
- [Layout](#)
- [loader](#)
- [LoadingItems](#)
- [macro](#)
- [Manifold](#)
- [ManifoldPoint](#)
- [Mask](#)
- [misc](#)
- [MotionStreak](#)
- [MotorJoint](#)
- [MouseJoint](#)
- [Node](#)
- [Node.EventType](#)
- [NodePool](#)
- [Object](#)
- [OriginalContainer](#)
- [PageView](#)
- [PageViewIndicator](#)
- [ParticleAsset](#)
- [ParticleSystem](#)
- [path](#)
- [PhysicsBoxCollider](#)
- [PhysicsChainCollider](#)
- [PhysicsCircleCollider](#)

- [PhysicsCollider](#)
- [PhysicsContact](#)
- [PhysicsImpulse](#)
- [PhysicsManager](#)
- [PhysicsPolygonCollider](#)
- [PhysicsRayCastResult](#)
- [Pipeline](#)
- [Pipeline.Downloader](#)
- [Pipeline.Loader](#)
- [Playable](#)
- [PolygonCollider](#)
- [Prefab](#)
- [PrismaticJoint](#)
- [PrivateNode](#)
- [ProgressBar](#)
- [ProportionalToFrame](#)
- [ProportionalToWindow](#)
- [Quat](#)
- [RawAsset](#)
- [Rect](#)
- [RenderComponent](#)
- [renderer](#)
- [RenderTexture](#)
- [ResolutionPolicy](#)
- [RevoluteJoint](#)
- [RichText](#)
- [RigidBody](#)
- [RopeJoint](#)
- [Scene](#)
- [SceneAsset](#)
- [Scheduler](#)
- [screen](#)
- [Scrollbar](#)
- [ScrollView](#)
- [Size](#)
- [Slider](#)
- [Sprite](#)
- [SpriteAtlas](#)
- [SpriteFrame](#)
- [sys](#)
- [SystemEvent](#)
- [SystemEvent.EventType](#)
- [TextAsset](#)
- [Texture2D](#)
- [TiledLayer](#)
- [TiledMap](#)

- [TiledMapAsset](#)
- [TiledObjectGroup](#)
- [TiledTile](#)
- [Toggle](#)
- [ToggleContainer](#)
- [ToggleGroup](#)
- [Touch](#)
- [TTFFont](#)
- [TypeScript](#)
- [url](#)
- [ValueType](#)
- [Vec2](#)
- [Vec3](#)
- [VideoPlayer](#)
- [View](#)
- [ViewGroup](#)
- [visibleRect](#)
- [WebView](#)
- [WeldJoint](#)
- [WheelJoint](#)
- [Widget](#)
- [WorldManifold](#)
- [WXSubContextView](#)

Enums

- [AccountGender](#)
- [AccountOperate](#)
- [AccountType](#)
- [AdsPos](#)
- [AdsresultCode](#)
- [AdsType](#)
- [AnimationEventType](#)
- [audioEngine.AudioState](#)
- [Button.Transition](#)
- [Camera.ClearFlags](#)
- [CustomresultCode](#)
- [debug.DebugMode](#)
- [DrawBits](#)
- [EditBox.InputFlag](#)
- [EditBox.InputMode](#)
- [EditBox.KeyboardReturnType](#)
- [Graphics.LineCap](#)
- [Graphics.LineJoin](#)
- [Label.HorizontalAlign](#)
- [Label.Overflow](#)

- [Label.Type](#)
- [Label.VerticalAlign](#)
- [Layout.AxisDirection](#)
- [Layout.HorizontalDirection](#)
- [Layout.ResizeMode](#)
- [Layout.Type](#)
- [Layout.VerticalDirection](#)
- [LoadingItems.ItemState](#)
- [macro.BlendFactor](#)
- [macro.ImageFormat](#)
- [macro.KEY](#)
- [macro.TextAlignment](#)
- [Mask.Type](#)
- [NetworkType](#)
- [PageView.Direction](#)
- [PageView.EventType](#)
- [PageView.SizeMode](#)
- [PageViewIndicator.Direction](#)
- [ParticleSystem.EmitterMode](#)
- [ParticleSystem.PositionType](#)
- [PayResultCode](#)
- [Prefab.OptimizationPolicy](#)
- [ProgressBar.Mode](#)
- [PushActionResultCode](#)
- [RayCastType](#)
- [RECResultCode](#)
- [RigidBodyType](#)
- [Scrollbar.Direction](#)
- [ScrollView.EventType](#)
- [ShareResultCode](#)
- [Slider.Direction](#)
- [SocialRetCode](#)
- [Sprite.FillType](#)
- [Sprite.SizeMode](#)
- [Sprite.Type](#)
- [TaskType](#)
- [Texture2D.Filter](#)
- [Texture2D.PixelFormat](#)
- [Texture2D.WrapMode](#)
- [TiledMap.Orientation](#)
- [ToolBarPlace](#)
- [UserActionResultCode](#)
- [VerticalTextAlignment](#)
- [VideoPlayer.EventType](#)
- [VideoPlayer.ResourceType](#)
- [WebView.EventType](#)

- [Widget.AlignMode](#)
- [WrapMode](#)

Index

Properties

- `ENGINE_VERSION` String The current version of Cocos2d being used....
- `director` Director Director
- `systemEvent` SystemEvent The System event singleton for global usage
- `view` View cc.view is the shared view object.
- `winSize` Size cc.winSize is the alias object for the size of the current game window.
- `Integer` string Specify that the input value must be integer in Inspector.
- `Float` string Indicates that the elements in array should be type double.
- `Boolean` string Indicates that the elements in array should be type boolean.
- `String` string Indicates that the elements in array should be type string.

Methods

- `speed` or less (speed < 1) time.
- `follow` Create a follow action which makes its target follows another node.
- `setPoints` Points setter
- `cardinalSplineTo` Creates an action with a Cardinal Spline array of points and tension.
- `updatePosition` update position of target
- `cardinalSplineBy` Creates an action with a Cardinal Spline array of points and tension.
- `catmullRomTo` Creates an action with a Cardinal Spline array of points and tension.
- `catmullRomBy` Creates an action with a Cardinal Spline array of points and tension.
- `easeIn` Creates the action easing object with the rate parameter.
- `easeOut` Creates the action easing object with the rate parameter.
- `easeInOut` Creates the action easing object with the rate parameter.
- `easeExponentialIn` Creates the action easing object with the rate parameter.
- `easeExponentialOut` Creates the action easing object.
- `easeExponentialInOut` Creates an EaseExponentialInOut action easing object.
- `easeSineIn` Creates an EaseSineIn action.
- `easeSineOut` Creates an EaseSineOut action easing object.
- `easeSineInOut` Creates the action easing object.
- `easeElasticIn` Creates the action easing object with the period in radians (default is 0.3).
- `easeElasticOut` Creates the action easing object with the period in radians (default is 0.3).
- `easeElasticInOut` Creates the action easing object with the period in radians (default is 0.3).
- `easeBounceIn` Creates the action easing object.
- `easeBounceOut` Creates the action easing object.
- `easeBounceInOut` Creates the action easing object.
- `easeBackIn` Creates the action easing object.

- `easeBackOut` Creates the action easing object.
- `easeBackInOut` Creates the action easing object.
- `easeBezierAction` Creates the action easing object.
- `easeQuadraticActionIn` Creates the action easing object.
- `easeQuadraticActionOut` Creates the action easing object.
- `easeQuadraticActionInOut` Creates the action easing object.
- `easeQuarticActionIn` Creates the action easing object.
- `easeQuarticActionOut` Creates the action easing object.
- `easeQuarticActionInOut` Creates the action easing object.
- `easeQuinticActionIn` Creates the action easing object.
- `easeQuinticActionOut` Creates the action easing object.
- `easeQuinticActionInOut` Creates the action easing object.
- `easeCircleActionIn` Creates the action easing object.
- `easeCircleActionOut` Creates the action easing object.
- `easeCircleActionInOut` Creates the action easing object.
- `easeCubicActionIn` Creates the action easing object.
- `easeCubicActionOut` Creates the action easing object.
- `easeCubicActionInOut` Creates the action easing object.
- `show` Show the Node.
- `hide` Hide the node.
- `toggleVisibility` Toggles the visibility of a node.
- `removeSelf` Create a RemoveSelf object with a flag indicate whether the target should be cleaned up while removing.
- `flipX` Create a FlipX action to flip or unflip the target.
- `flipY` Create a FlipY action to flip or unflip the target.
- `place` Creates a Place action with a position.
- `callFunc` Creates the action with the callback.
- `sequence` Helper constructor to create an array of sequenceable actions
- `repeat` Creates a Repeat action.
- `repeatForever` Create a action which repeat forever, as it runs forever, it can't be added into cc.sequence and cc.spawn.
- `spawn` Create a spawn action which runs several actions in parallel.
- `rotateTo` Rotates a Node object to a certain angle by modifying its rotation property.
- `rotateBy` Rotates a Node object clockwise a number of degrees by modifying its rotation property.
- `moveBy` Moves a Node object x,y pixels by modifying its position property.
- `moveTo` Moves a Node object to the position x,y.
- `skewTo` Create a action which skews a Node object to given angles by modifying its skewX and skewY properties.
- `skewBy` Skews a Node object by skewX and skewY degrees.
- `jumpBy` Moves a Node object simulating a parabolic jump movement by modifying it's position property.
- `jumpTo` Moves a Node object to a parabolic position simulating a jump movement by modifying its position property.
- `bezierBy` An action that moves the target with a cubic Bezier curve by a certain distance.

- `bezierTo` An action that moves the target with a cubic Bezier curve to a destination point.
- `scaleTo` Scales a Node object to a zoom factor by modifying it's scale property.
- `scaleBy` Scales a Node object a zoom factor by modifying it's scale property.
- `blink` Blinks a Node object by modifying it's visible property.
- `fadeTo` Fades an object that implements the cc.RGBAProtocol protocol.
- `fadeIn` Fades In an object that implements the cc.RGBAProtocol protocol.
- `fadeOut` Fades Out an object that implements the cc.RGBAProtocol protocol.
- `tintTo` Tints a Node that implements the cc.NodeRGB protocol from current tint to a custom one.
- `tintBy` Tints a Node that implements the cc.NodeRGB protocol from current tint to a custom one.
- `delayTime` Delays the action a certain amount of seconds.
- `reverseTime` Executes an action in reverse order, from time=duration to time=0.
- `targetedAction` Create an action with the specified action and forced target.
- `error` Outputs an error message to the Cocos Creator Console (editor) or Web Console (runtime)....
- `warn` Outputs a warning message to the Cocos Creator Console (editor) or Web Console (runtime).
- `log` Outputs a message to the Cocos Creator Console (editor) or Web Console (runtime).
- `find` Finds a node by hierarchy path, the path is case-sensitive.
- `color` Alpha channel is optional.
- `rect` The convenience method to create a new Rect.
- `size` Helper function that creates a cc.Size....
- `v2` The convenience method to create a new [cc.Vec2](#).
- `p` This function is deprecated since v2.0, please use V2.
- `Class` Defines a CCClass using the given specification, please see [Class](#) for details.
- `_isCCClass` Checks whether the constructor is created by cc.Class
- `Enum` Define an enum type.
- `getList`
- `handleTouchesBegin`
- `handleTouchesMove`
- `handleTouchesEnd`
- `handleTouchesCancel`
- `getSetOfTouchesEndOrCancel`
- `getHTMLElementPosition`
- `getPreTouch`
- `setPreTouch`
- `getTouchByXY`
- `getMouseEvent`
- `getPointByEvent`
- `getTouchesByEvent`
- `registerSystemEvent`
- `update`
- `isValid` When an object's destroy is called, it is actually destroyed after the end of this frame.
- `deserialize` Deserialize json to cc.Asset

- `instantiate` Clones the object `original` and returns the clone, or instantiate a node from the Prefab.

Details

Properties

ENGINE_VERSION

The current version of Cocos2d being used.

Please DO NOT remove this String, it is an important flag for bug tracking.
If you post a bug to forum, please attach this flag.

meta	description
Type	String
Defined in	predefine.js:160

director

Director

meta	description
Type	Director
Defined in	cocos2d/core/CCDirector.js:1157

systemEvent

The System event singleton for global usage

meta	description
Type	SystemEvent
Defined in	cocos2d/core/event/system-event.js:173

view

cc.view is the shared view object.

meta	description
Type	View
Defined in	cocos2d/core/platform/CCView.js:1526

winSize

cc.winSize is the alias object for the size of the current game window.

meta	description
Type	Size
Defined in	cocos2d/core/platform/CCView.js:1535

Integer

Specify that the input value must be integer in Inspector. Also used to indicates that the elements in array should be type integer.

meta	description
Type	string
Defined in	cocos2d/core/platform/attribute.js:153

Examples

```
// in cc.Class
member: {
    default: [],
    type: cc.Integer
}
// ES6 ccclass
```

Float

Indicates that the elements in array should be type double.

meta	description
Type	string
Defined in	cocos2d/core/platform/attribute.js:172

Examples

```
// in cc.Class
member: {
    default: [],
    type: cc.Float
}
// ES6 ccclass
```

Boolean

Indicates that the elements in array should be type boolean.

meta	description
Type	string
Defined in	cocos2d/core/platform/attribute.js:197

Examples

```
// in cc.Class
member: {
    default: [],
    type: cc.Boolean
}
// ES6 ccclass
```

String

Indicates that the elements in array should be type string.

meta	description
Type	string
Defined in	cocos2d/core/platform/attribute.js:215

Examples

```
// in cc.Class
member: {
    default: [],
    type: cc.String
}
// ES6 ccclass
```

Methods

speed

Creates the speed action which changes the speed of an action, making it take longer (speed > 1) or less (speed < 1) time.

Useful to simulate 'slow motion' or 'fast forward' effect.

meta	description
Returns	Action
Defined in	cocos2d/actions/CCAction.js:372

Parameters

- action [ActionInterval](#)
- speed [Number](#)

Examples

```
// change the target action speed;
var action = cc.scaleTo(0.2, 1, 0.6);
var newAction = cc.speed(action, 0.5);
```

follow

Create a follow action which makes its target follows another node.

meta	description
Returns	Action Null
Defined in	cocos2d/actions/CCAction.js:556

Parameters

- followedNode [Node](#)
- rect [Rect](#)

Examples

```
// example
// creates the action with a set boundary
var followAction = cc.follow(targetNode, cc.rect(0, 0, screenWidth * 2 - 100,
screenHeight));
node.runAction(followAction);

// creates the action with no boundary set
var followAction = cc.follow(targetNode);
node.runAction(followAction);
```

setPoints

Points setter

meta	description
Defined in	cocos2d/actions/CCActionCatmullRom.js:226

Parameters

- points [Array](#)

cardinalSplineTo

Creates an action with a Cardinal Spline array of points and tension.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionCatmullRom.js:236

Parameters

- duration [Number](#)
- points [Array](#) array of control points
- tension [Number](#)

Examples

```
//create a cc.CardinalSplineTo  
var action1 = cc.cardinalSplineTo(3, array, 0);
```

updatePosition

update position of target

meta	description
Defined in	cocos2d/actions/CCActionCatmullRom.js:319

Parameters

- newPos [Vec2](#)

cardinalSplineBy

Creates an action with a Cardinal Spline array of points and tension.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionCatmullRom.js:340

Parameters

- duration [Number](#)
- points [Array](#)
- tension [Number](#)

catmullRomTo

Creates an action with a Cardinal Spline array of points and tension.

meta	description
Returns	ActionInterval

meta	description
Defined in	cocos2d/actions/CCActionCatmullRom.js:388

Parameters

- dt [Number](#)
- points [Array](#)

Examples

```
var action1 = cc.catmullRomTo(3, array);
```

catmullRomBy

Creates an action with a Cardinal Spline array of points and tension.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionCatmullRom.js:437

Parameters

- dt [Number](#)
- points [Array](#)

Examples

```
var action1 = cc.catmullRomBy(3, array);
```

easeIn

Creates the action easing object with the rate parameter.
From slow to fast.

meta	description
Returns	Object

meta	description
Defined in	cocos2d/actions/CCActionEase.js:35

Parameters

- `rate` [Number](#)

Examples

```
action.easing(cc.easeIn(3.0));
```

easeOut

Creates the action easing object with the rate parameter.
From fast to slow.

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:58

Parameters

- `rate` [Number](#)

Examples

```
action.easing(cc.easeOut(3.0));
```

easeInOut

Creates the action easing object with the rate parameter.
Slow to fast then to slow.

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:81

Parameters

- `rate` [Number](#)

Examples

```
action.easing(cc.easeInOut(3.0));
```

easeExponentialIn

Creates the action easing object with the rate parameter.

Reference easeInExpo:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:109

Examples

```
action.easing(cc.easeExponentialIn());
```

easeExponentialOut

Creates the action easing object.

Reference easeOutExpo:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:135

Examples

```
action.easing(cc.easeExponentialOut());
```

easeExponentialInOut

Creates an EaseExponentialInOut action easing object.

Reference easeInOutExpo:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:161

Examples

```
action.easing(cc.easeExponentialInOut());
```

easeSineIn

Creates an EaseSineIn action.

Reference easeInSine:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:194

Examples

```
action.easing(cc.easeSineIn());
```

easeSineOut

Creates an EaseSineOut action easing object.

Reference easeOutSine:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:220

Examples

```
action.easing(cc.easeSineOut());
```

easeSineInOut

Creates the action easing object.

Reference easeInOutSine:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:246

Examples

```
action.easing(cc.easeSineInOut());
```

easeElasticIn

Creates the action easing object with the period in radians (default is 0.3).

Reference easeInElastic:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:276

Parameters

- period [Number](#)

Examples

```
// example
action.easing(cc.easeElasticIn(3.0));
```

easeElasticOut

Creates the action easing object with the period in radians (default is 0.3).
Reference easeOutElastic:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:322

Parameters

- period [Number](#)

Examples

```
// example
action.easing(cc.easeElasticOut(3.0));
```

easeElasticInOut

Creates the action easing object with the period in radians (default is 0.3).
Reference easeInOutElastic:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:362

Parameters

- period [Number](#)

Examples

```
// example
action.easing(cc.easeElasticInOut(3.0));
```

easeBounceIn

Creates the action easing object.
Eased bounce effect at the beginning.

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:434

Examples

```
// example
action.easing(cc.easeBounceIn());
```

easeBounceOut

Creates the action easing object.
Eased bounce effect at the ending.

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:451

Examples

```
// example
action.easing(cc.easeBounceOut());
```

easeBounceInOut

Creates the action easing object.
Eased bounce effect at the begining and ending.

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:476

Examples

```
// example
action.easing(cc.easeBounceInOut());
```

easeBackIn

Creates the action easing object.

In the opposite direction to move slowly, and then accelerated to the right direction.

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:508

Examples

```
// example
action.easing(cc.easeBackIn());
```

easeBackOut

Creates the action easing object.

Fast moving more than the finish, and then slowly back to the finish.

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:534

Examples

```
// example
action.easing(cc.easeBackOut());
```

easeBackInOut

Creates the action easing object.

Begining of cc.EaseBackIn. Ending of cc.EaseBackOut.

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:561

Examples

```
// example
action.easing(cc.easeBackInOut());
```

easeBezierAction

Creates the action easing object.
 Into the 4 reference point.
 To calculate the motion curve.

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:592

Parameters

- `p0 Number` The first bezier parameter
- `p1 Number` The second bezier parameter
- `p2 Number` The third bezier parameter
- `p3 Number` The fourth bezier parameter

Examples

```
// example
action.easing(cc.easeBezierAction(0.5, 0.5, 1.0, 1.0));
```

easeQuadraticActionIn

Creates the action easing object.
 Reference easeInQuad:
<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:621

Examples

```
//example
action.easing(cc.easeQuadraticActionIn());
```

easeQuadraticActionOut

Creates the action easing object.

Reference easeOutQuad:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:648

Examples

```
//example
action.easing(cc.easeQuadraticActionOut());
```

easeQuadraticActionInOut

Creates the action easing object.

Reference easeInOutQuad:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:675

Examples

```
//example
action.easing(cc.easeQuadraticActionInOut());
```

easeQuarticActionIn

Creates the action easing object.

Reference easeInQuart:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:710

Examples

```
//example
action.easing(cc.easeQuarticActionIn());
```

easeQuarticActionOut

Creates the action easing object.

Reference easeOutQuart:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:737

Examples

```
//example
action.easing(cc.QuarticActionOut());
```

easeQuarticActionInOut

Creates the action easing object.

Reference easeInOutQuart:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:765

easeQuinticActionIn

Creates the action easing object.

Reference easeInQuint:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:793

Examples

```
//example
action.easing(cc.easeQuinticActionIn());
```

easeQuinticActionOut

Creates the action easing object.

Reference easeOutQuint:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:820

Examples

```
//example
action.easing(cc.easeQuadraticActionOut());
```

easeQuinticActionInOut

Creates the action easing object.

Reference easeInOutQuint:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:848

Examples

```
//example
action.easing(cc.easeQuinticActionInOut());
```

easeCircleActionIn

Creates the action easing object.

Reference easeInCirc:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:879

Examples

```
//example
action.easing(cc.easeCircleActionIn());
```

easeCircleActionOut

Creates the action easing object.

Reference easeOutCirc:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object

meta	description
Defined in	cocos2d/actions/CCActionEase.js:906

Examples

```
//example
action.easing(cc.easeCircleActionOut());
```

easeCircleActionInOut

Creates the action easing object.

Reference easeInOutCirc:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:934

Examples

```
//example
action.easing(cc.easeCircleActionInOut());
```

easeCubicActionIn

Creates the action easing object.

Reference easeInCubic:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:965

Examples

```
//example
action.easing(cc.easeCubicActionIn());
```

easeCubicActionOut

Creates the action easing object.

Reference easeOutCubic:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:992

Examples

```
//example
action.easing(cc.easeCubicActionOut());
```

easeCubicActionInOut

Creates the action easing object.

Reference easeInOutCubic:

<http://www.zhihu.com/question/21981571/answer/19925418>

meta	description
Returns	Object
Defined in	cocos2d/actions/CCActionEase.js:1020

show

Show the Node.

meta	description
Returns	ActionInstant
Defined in	cocos2d/actions/CCActionInstant.js:102

Examples

```
// example
var showAction = cc.show();
```

hide

Hide the node.

meta	description
Returns	ActionInstant
Defined in	cocos2d/actions/CCActionInstant.js:141

Examples

```
// example
var hideAction = cc.hide();
```

toggleVisibility

Toggles the visibility of a node.

meta	description
Returns	ActionInstant
Defined in	cocos2d/actions/CCActionInstant.js:180

Examples

```
// example
var toggleVisibilityAction = cc.toggleVisibility();
```

removeSelf

Create a RemoveSelf object with a flag indicate whether the target should be cleaned up while removing.

meta	description
Returns	ActionInstant

meta	description
Defined in	cocos2d/actions/CCActionInstant.js:230

Parameters

- `isNeedCleanUp Boolean`

Examples

```
// example
var removeSelfAction = cc.removeSelf();
```

flipX

Create a FlipX action to flip or unflip the target.

meta	description
Returns	ActionInstant
Defined in	cocos2d/actions/CCActionInstant.js:288

Parameters

- `flip Boolean` Indicate whether the target should be flipped or not

Examples

```
var flipXAction = cc.flipX(true);
```

flipY

Create a FlipY action to flip or unflip the target.

meta	description
Returns	ActionInstant
Defined in	cocos2d/actions/CCActionInstant.js:343

Parameters

- `flip` [Boolean](#)

Examples

```
var flipYAction = cc.flipY(true);
```

place

Creates a Place action with a position.

meta	description
Returns	ActionInstant
Defined in	cocos2d/actions/CCActionInstant.js:406

Parameters

- `pos` [Vec2](#) | [Number](#)
- `y` [Number](#)

Examples

```
// example
var placeAction = cc.place(cc.v2(200, 200));
var placeAction = cc.place(200, 200);
```

callFunc

Creates the action with the callback.

meta	description
Returns	ActionInstant
Defined in	cocos2d/actions/CCActionInstant.js:516

Parameters

- `selector` [function](#)
- `selectorTarget` [object](#)

- `data` Any data for function, it accepts all data types.

Examples

```
// example
// CallFunc without data
var finish = cc.callFunc(this.removeSprite, this);

// CallFunc with data
var finish = cc.callFunc(this.removeFromParentAndCleanup, this._grossini, true);
```

sequence

Helper constructor to create an array of sequenceable actions The created action will run actions sequentially, one after another.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:438

Parameters

- `actionOrActionArray` [FiniteTimeAction](#) | [FiniteTimeAction\[\]](#)
- `tempArray` [FiniteTimeAction](#)

Examples

```
// example
// create sequence with actions
var seq = cc.sequence(act1, act2);

// create sequence with array
var seq = cc.sequence(actArray);
```

repeat

Creates a Repeat action. Times is an unsigned integer between 1 and pow(2,30)

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:616

Parameters

- `action` [FiniteTimeAction](#)
- `times` [Number](#)

Examples

```
// example
var rep = cc.repeat(cc.sequence(jump2, jump1), 5);
```

repeatForever

Create a action which repeat forever, as it runs forever, it can't be added into cc.sequence and cc.spawn.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:720

Parameters

- `action` [FiniteTimeAction](#)

Examples

```
// example
var repeat = cc.repeatForever(cc.rotateBy(1.0, 360));
```

spawn

Create a spawn action which runs several actions in parallel.

meta	description
Returns	FiniteTimeAction
Defined in	cocos2d/actions/CCActionInterval.js:835

Parameters

- `actionOrActionArray` [FiniteTimeAction](#) | [FiniteTimeAction\[\]](#)

- `tempArray` [FiniteTimeAction](#)

Examples

```
// example
var action = cc.spawn(cc.jumpBy(2, cc.v2(300, 0), 50, 4), cc.rotateBy(2, 720));
todo:It should be the direct use new
```

rotateTo

Rotates a Node object to a certain angle by modifying its rotation property. The direction will be decided by the shortest angle.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:955

Parameters

- `duration` [Number](#) duration in seconds
- `deltaAngleX` [Number](#) deltaAngleX in degrees.
- `deltaAngleY` [Number](#) deltaAngleY in degrees.

Examples

```
// example
var rotateTo = cc.rotateTo(2, 61.0);
```

rotateBy

Rotates a Node object clockwise a number of degrees by modifying its rotation property. Relative to its properties to modify.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:1042

Parameters

- duration [Number](#) duration in seconds
- deltaAngleX [Number](#) deltaAngleX in degrees
- deltaAngleY [Number](#) deltaAngleY in degrees

Examples

```
// example
var actionBy = cc.rotateBy(2, 360);
```

moveBy

Moves a Node object x,y pixels by modifying its position property.
x and y are relative to the position of the object.
Several MoveBy actions can be concurrently called, and the resulting movement will be the sum of individual movements.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:1158

Parameters

- duration [Number](#) duration in seconds
- deltaPos [Vec2](#) | [Number](#)
- deltaY [Number](#)

Examples

```
// example
var actionTo = cc.moveTo(2, cc.v2(windowSize.width - 40, windowSize.height - 40));
```

moveTo

Moves a Node object to the position x,y. x and y are absolute coordinates by modifying its position property.
Several MoveTo actions can be concurrently called, and the resulting movement will be the sum of individual movements.

meta	description
Returns	ActionInterval

meta	description
Defined in	cocos2d/actions/CCActionInterval.js:1235

Parameters

- duration [Number](#) duration in seconds
- position [Vec2](#) | [Number](#)
- y [Number](#)

Examples

```
// example
var actionBy = cc.moveTo(2, cc.v2(80, 80));
```

skewTo

Create a action which skews a Node object to given angles by modifying its skewX and skewY properties. Changes to the specified value.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:1329

Parameters

- t [Number](#) time in seconds
- sx [Number](#)
- sy [Number](#)

Examples

```
// example
var actionTo = cc.skewTo(2, 37.2, -37.2);
```

skewBy

Skews a Node object by skewX and skewY degrees.
Relative to its property modification.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:1404

Parameters

- t [Number](#) time in seconds
- sx [Number](#) sx skew in degrees for X axis
- sy [Number](#) sy skew in degrees for Y axis

Examples

```
// example
var actionBy = cc.skewBy(2, 0, -90);
```

jumpBy

Moves a Node object simulating a parabolic jump movement by modifying its position property. Relative to its movement.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:1531

Parameters

- duration [Number](#)
- position [Vec2](#) | [Number](#)
- y [Number](#)
- height [Number](#)
- jumps [Number](#)

Examples

```
// example
var actionBy = cc.jumpBy(2, cc.v2(300, 0), 50, 4);
var actionBy = cc.jumpBy(2, 300, 0, 50, 4);
```

jumpTo

Moves a Node object to a parabolic position simulating a jump movement by modifying its position property.
Jump to the specified location.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:1613

Parameters

- duration [Number](#)
- position [Vec2](#) | [Number](#)
- y [Number](#)
- height [Number](#)
- jumps [Number](#)

Examples

```
// example
var actionTo = cc.jumpTo(2, cc.v2(300, 300), 50, 4);
var actionTo = cc.jumpTo(2, 300, 300, 50, 4);
```

bezierBy

An action that moves the target with a cubic Bezier curve by a certain distance.
Relative to its movement.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:1749

Parameters

- t [Number](#) time in seconds
- c [Vec2\[\]](#) Array of points

Examples

```
// example
```

```
var bezier = [cc.v2(0, windowHeight.height / 2), cc.v2(300, -windowHeight.height / 2),
cc.v2(300, 100)];
var bezierForward = cc.bezierBy(3, bezier);
```

bezierTo

An action that moves the target with a cubic Bezier curve to a destination point.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:1818

Parameters

- t [Number](#)
- c [Vec2\[\]](#) Array of points

Examples

```
// example
var bezier = [cc.v2(0, windowHeight.height / 2), cc.v2(300, -windowHeight.height / 2),
cc.v2(300, 100)];
var bezierTo = cc.bezierTo(2, bezier);
```

scaleTo

Scales a Node object to a zoom factor by modifying it's scale property.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:1904

Parameters

- duration [Number](#)
- sx [Number](#) scale parameter in X
- sy [Number](#) scale parameter in Y, if Null equal to sx

Examples

```
// example
// It scales to 0.5 in both X and Y.
var actionTo = cc.scaleTo(2, 0.5);

// It scales to 0.5 in x and 2 in Y
var actionTo = cc.scaleTo(2, 0.5, 2);
```

scaleBy

Scales a Node object a zoom factor by modifying it's scale property. Relative to its changes.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:1954

Parameters

- duration [Number](#) duration in seconds
- sx [Number](#) sx scale parameter in X
- sy [Number](#) | Null sy scale parameter in Y, if Null equal to sx

Examples

```
// example without sy, it scales by 2 both in X and Y
var actionBy = cc.scaleBy(2, 2);

//example with sy, it scales by 0.25 in X and 4.5 in Y
var actionBy2 = cc.scaleBy(2, 0.25, 4.5);
```

blink

Blinks a Node object by modifying it's visible property.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:2040

Parameters

- duration [Number](#) duration in seconds
- blinks [Number](#) blinks in times

Examples

```
// example
var action = cc.blink(2, 10);
```

fadeTo

Fades an object that implements the cc.RGBAProtocol protocol. It modifies the opacity from the current value to a custom one.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:2107

Parameters

- duration [Number](#)
- opacity [Number](#) 0-255, 0 is transparent

Examples

```
// example
var action = cc.fadeTo(1.0, 0);
```

fadeIn

Fades In an object that implements the cc.RGBAProtocol protocol. It modifies the opacity from 0 to 255.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:2163

Parameters

- duration [Number](#) duration in seconds

Examples

```
//example
var action = cc.fadeIn(1.0);
```

fadeOut

Fades Out an object that implements the cc.RGBAProtocol protocol. It modifies the opacity from 255 to 0.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:2212

Parameters

- d [Number](#) duration in seconds

Examples

```
// example
var action = cc.fadeOut(1.0);
```

tintTo

Tints a Node that implements the cc.NodeRGB protocol from current tint to a custom one.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:2296

Parameters

- duration [Number](#)
- red [Number](#) 0-255
- green [Number](#) 0-255
- blue [Number](#) 0-255

Examples

```
// example
var action = cc.tintTo(2, 255, 0, 255);
```

tintBy

Tints a Node that implements the cc.NodeRGB protocol from current tint to a custom one. Relative to their own color change.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:2389

Parameters

- duration [Number](#) duration in seconds
- deltaRed [Number](#)
- deltaGreen [Number](#)
- deltaBlue [Number](#)

Examples

```
// example
var action = cc.tintBy(2, -127, -255, -127);
```

delayTime

Delays the action a certain amount of seconds. 延迟指定的时间量。

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:2433

Parameters

- d [Number](#) duration in seconds

Examples

```
// example
var delay = cc.delayTime(1);
```

reverseTime

Executes an action in reverse order, from time=duration to time=0.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:2519

Parameters

- `action` [FiniteTimeAction](#)

Examples

```
// example
var reverse = cc.reverseTime(this);
```

targetedAction

Create an action with the specified action and forced target.

meta	description
Returns	ActionInterval
Defined in	cocos2d/actions/CCActionInterval.js:2607

Parameters

- `target` [Node](#)
- `action` [FiniteTimeAction](#)

error

Outputs an error message to the Cocos Creator Console (editor) or Web Console (runtime).

- In Cocos Creator, error is red.

- In Chrome, error have a red icon along with red message text.

meta	description
Defined in	cocos2d/core/CCDebug.js:111

Parameters

- `msg` Any A JavaScript string containing zero or more substitution strings.
- `subst` Any JavaScript objects with which to replace substitution strings within `msg`. This gives you additional control over the format of the output.

warn

Outputs a warning message to the Cocos Creator Console (editor) or Web Console (runtime).

- In Cocos Creator, warning is yellow.
- In Chrome, warning have a yellow warning icon with the message text.

meta	description
Defined in	cocos2d/core/CCDebug.js:155

Parameters

- `msg` Any A JavaScript string containing zero or more substitution strings.
- `subst` Any JavaScript objects with which to replace substitution strings within `msg`. This gives you additional control over the format of the output.

log

Outputs a message to the Cocos Creator Console (editor) or Web Console (runtime).

meta	description
Defined in	cocos2d/core/CCDebug.js:185

Parameters

- `msg` [String](#) | Any A JavaScript string containing zero or more substitution strings.

- `subst` Any JavaScript objects with which to replace substitution strings within msg. This gives you additional control over the format of the output.

find

Finds a node by hierarchy path, the path is case-sensitive. It will traverse the hierarchy by splitting the path using '/' character. This function will still returns the node even if it is inactive. It is recommended to not use this function every frame instead cache the result at startup.

meta	description
Returns	Node Null
Defined in	cocos2d/core/utils/find.js:30

Parameters

- `path` [String](#)
- `referenceNode` [Node](#)

color

The convenience method to create a new Color/Color:method Alpha channel is optional. Default value is 255.

meta	description
Returns	Color
Defined in	cocos2d/core/value-types/color.js:588

Parameters

- `r` [Number](#)
- `g` [Number](#)
- `b` [Number](#)
- `a` [Number](#)

Examples

```Not found for the example path: temp-src/engine/docs/utils/api/engine/docs/cocos2d/core/value-types/CCColor/color.js

## rect

The convenience method to create a new Rect. see Rect/Rect:method

| meta       | description                                          |
|------------|------------------------------------------------------|
| Returns    | <a href="#">Rect</a>                                 |
| Defined in | <a href="#">cocos2d/core/value-types/rect.js:423</a> |

### Parameters

- `x` [Number](#)
- `y` [Number](#)
- `w` [Number](#)
- `h` [Number](#)

## Examples

```
var a = new cc.Rect(0 , 0, 10, 0);
```

## size

Helper function that creates a cc.Size.

Please use cc.p or cc.v2 instead, it will soon replace cc.Size.

| meta       | description                                          |
|------------|------------------------------------------------------|
| Returns    | <a href="#">Size</a>                                 |
| Defined in | <a href="#">cocos2d/core/value-types/size.js:163</a> |

### Parameters

- `w` [Number](#) | [Size](#) width or a size object
- `h` [Number](#) height

## Examples

```Not found for the example path: temp-src/engine/docs/utils/api/engine/docs/cocos2d/core/value-types/CCSize/size.js

v2

The convenience method to create a new [cc.Vec2](#).

| meta | description |
|------------|--|
| Returns | Vec2 |
| Defined in | cocos2d/core/value-types/vec2.js:646 |

Parameters

- x [Number](#) | [Object](#)
- y [Number](#)

Examples

```
var v1 = cc.v2();
var v2 = cc.v2(0, 0);
var v3 = cc.v2(v2);
var v4 = cc.v2({x: 100, y: 100});
```

p

This function is deprecated since v2.0, please use V2.

| meta | description |
|------------|--|
| Returns | Vec2 |
| Defined in | cocos2d/core/value-types/vec2.js:663 |
| Deprecated | since v2.0 |

Parameters

- x [Number](#) | [Object](#) a Number or a size object
- y [Number](#)

Class

Defines a CCClass using the given specification, please see [Class](#) for details.

| meta | description |
|------------|--|
| Returns | Function |
| Defined in | cocos2d/core/platform/CCClass.js:846 |

Parameters

- options [Object](#)
- name [String](#) The class name used for serialization.
- extends [Function](#) The base class.
- ctor [Function](#) The constructor.
- __ctor__ [Function](#) The same as ctor, but less encapsulated.
- properties [Object](#) The property definitions.
- statics [Object](#) The static members.
- mixins [Function\[\]](#)
- editor [Object](#) attributes for Component listed below.
- executeInEditMode [Boolean](#) Allows the current component to run in edit mode. By default, all components are executed only at runtime, meaning that they will not have their callback functions executed while the Editor is in edit mode.
- requireComponent [Function](#) Automatically add required component as a dependency.
- menu [String](#) The menu path to register a component to the editors "Component" menu. Eg. "Rendering/Camera".
- executionOrder [Number](#) The execution order of lifecycle methods for Component. Those less than 0 will execute before while those greater than 0 will execute after. The order will only affect onLoad, onEnable, start, update and lateUpdate while onDisable and onDestroy will not be affected.
- disallowMultiple [Boolean](#) If specified to a type, prevents Component of the same type (or subtype) to be added more than once to a Node.
- playOnFocus [Boolean](#) This property is only available when executeInEditMode is set. If specified, the editor's scene view will keep updating this node in 60 fps when it is selected, otherwise, it will update only if necessary.
- inspector [String](#) Customize the page url used by the current component to render in the Properties.
- icon [String](#) Customize the icon that the current component displays in the editor.
- help [String](#) The custom documentation URL
- update [Function](#) lifecycle method for Component, see [update](#)
- lateUpdate [Function](#) lifecycle method for Component, see [lateUpdate](#)
- onLoad [Function](#) lifecycle method for Component, see [onLoad](#)
- start [Function](#) lifecycle method for Component, see [start](#)

- `onEnable` [Function](#) lifecycle method for Component, see [onEnable](#)
- `onDisable` [Function](#) lifecycle method for Component, see [onDisable](#)
- `onDestroy` [Function](#) lifecycle method for Component, see [onDestroy](#)
- `onFocusInEditor` [Function](#) lifecycle method for Component, see [onFocusInEditor](#)
- `onLostFocusInEditor` [Function](#) lifecycle method for Component, see [onLostFocusInEditor](#)
- `resetInEditor` [Function](#) lifecycle method for Component, see [resetInEditor](#)
- `onRestore` [Function](#) for Component only, see [onRestore](#)
- `_getLocalBounds` [Function](#) for Component only, see [_getLocalBounds](#)

Examples

```
// define base class
var Node = cc.Class();

// define sub class
var Sprite = cc.Class({
name: 'Sprite',
extends: Node,

ctor: function () {
this.url = "";
this.id = 0;
},

statics: {
// define static members
count: 0,
getBounds: function (spriteList) {
// compute bounds...
}
},
properties {
width: {
default: 128,
type: 'Integer',
tooltip: 'The width of sprite'
},
height: 128,
size: {
get: function () {
return cc.v2(this.width, this.height);
}
}
},
load: function () {
// load this.url...
};
});

// instantiate
```

```
var obj = new Sprite();
obj.url = 'sprite.png';
obj.load();
```

_isCCClass

Checks whether the constructor is created by cc.Class

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/CCClass.js:1017 |

Parameters

- constructor [Function](#)

Enum

Define an enum type.

If a enum item has a value of -1, it will be given an Integer number according to its order in the list.

Otherwise it will use the value specified by user who writes the enum definition.

| meta | description |
|------------|--|
| Returns | object |
| Defined in | cocos2d/core/platform/CCEnum.js:34 |

Parameters

- obj [object](#) a JavaScript literal object containing enum names and values, or a TypeScript enum type

Examples

```
// JavaScript:

var WrapMode = cc.Enum({
    Repeat: -1,
    Clamp: -1
```

```

});

// Texture.WrapMode.Repeat == 0
// Texture.WrapMode.Clamp == 1
// Texture.WrapMode[0] == "Repeat"
// Texture.WrapMode[1] == "Clamp"

var FlagType = cc.Enum({
    Flag1: 1,
    Flag2: 2,
    Flag3: 4,
    Flag4: 8,
});

var AtlasSizeList = cc.Enum({
    128: 128,
    256: 256,
    512: 512,
    1024: 1024,
});

// TypeScript:

// If used in TypeScript, just define a TypeScript enum:
enum Direction {
    Up,
    Down,
    Left,
    Right
}

// If you need to inspect the enum in Properties panel, you can call cc.Enum:
const {ccclass, property} = cc._decorator;

@ccclass
class NewScript extends cc.Component {
    @property({
        type: cc.Enum(Direction)      // call cc.Enum
    })
    direction: Direction = Direction.Up;
}

```

getList

| meta | description |
|------------|--|
| Returns | Object[] |
| Defined in | cocos2d/core/platform/CCEnum.js:90 |

Parameters

- enumDef [Object](#) the enum type defined from cc.Enum

handleTouchesBegin

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCInputManager.js:100 |

Parameters

- touches [Array](#)

handleTouchesMove

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCInputManager.js:135 |

Parameters

- touches [Array](#)

handleTouchesEnd

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCInputManager.js:167 |

Parameters

- touches [Array](#)

handleTouchesCancel

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCInputManager.js:181 |

Parameters

- `touches` [Array](#)

getSetOfTouchesEndOrCancel

| meta | description |
|------------|---|
| Returns | Array |
| Defined in | cocos2d/core/platform/CCInputManager.js:195 |

Parameters

- `touches` [Array](#)

getHTMLElementPosition

| meta | description |
|------------|---|
| Returns | Object |
| Defined in | cocos2d/core/platform/CCInputManager.js:221 |

Parameters

- `element` [HTMLElement](#)

getPreTouch

| meta | description |
|------------|---|
| Returns | Touch |
| Defined in | cocos2d/core/platform/CCInputManager.js:268 |

Parameters

- `touch` [Touch](#)

setPreTouch

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCInputManager.js:288 |

Parameters

- touch [Touch](#)

getTouchByXY

| meta | description |
|------------|---|
| Returns | Touch |
| Defined in | cocos2d/core/platform/CCInputManager.js:313 |

Parameters

- tx [Number](#)
- ty [Number](#)
- pos [Vec2](#)

getMouseEvent

| meta | description |
|------------|---|
| Returns | Event.EventMouse |
| Defined in | cocos2d/core/platform/CCInputManager.js:330 |

Parameters

- location [Vec2](#)
- pos [Vec2](#)
- eventType [Number](#)

getPointByEvent

| meta | description |
|------------|---|
| Returns | Vec2 |
| Defined in | cocos2d/core/platform/CCInputManager.js:348 |

Parameters

- event [Touch](#)
- pos [Vec2](#)

getTouchesByEvent

| meta | description |
|------------|---|
| Returns | Array |
| Defined in | cocos2d/core/platform/CCInputManager.js:369 |

Parameters

- event [Touch](#)
- pos [Vec2](#)

registerSystemEvent

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCInputManager.js:407 |

Parameters

- element [HTMLElement](#)

update

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCInputManager.js:598 |

Parameters

- `dt Number`

isValid

Checks whether the object is non-nil and not yet destroyed.

When an object's `destroy` is called, it is actually destroyed after the end of this frame. So `isValid` will return false from the next frame, while `isValid` in the current frame will still be true. If you want to determine whether the current frame has called `destroy`, use `cc.isValid(obj, true)`, but this is often caused by a particular logical requirements, which is not normally required.

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:511 |

Parameters

- `value Any`
- `strictMode Boolean` If true, Object called `destroy()` in this frame will also treated as invalid.

Examples

```
var node = new cc.Node();
cc.log(cc.isValid(node));      // true
node.destroy();
cc.log(cc.isValid(node));      // true, still valid in this frame
// after a frame...
cc.log(cc.isValid(node));      // false, destroyed in the end of last frame
```

deserialize

Deserialize json to cc.Asset

| meta | description |
|------------|--|
| Returns | object |
| Defined in | cocos2d/core/platform/deserialize.js:760 |

Parameters

- data [String | Object](#) the serialized cc.Asset json string or json object.
- details [Details](#) additional loading result
- options [Object](#)

instantiate

Clones the object `original` and returns the clone, or instantiate a node from the Prefab.

| meta | description |
|------------|---|
| Returns | Node Object |
| Defined in | cocos2d/core/platform/instantiate.js:36 |

Parameters

- original [Prefab | Node | Object](#) An existing object that you want to make a copy of.

Examples

```
// instantiate node from prefab
var scene = cc.director.getScene();
var node = cc.instantiate(prefabAsset);
node.parent = scene;
// clone node
var scene = cc.director.getScene();
var node = cc.instantiate(targetNode);
node.parent = scene;
```

dragonBones Module

The global main namespace of DragonBones, all classes, functions, properties and constants of DragonBones are defined in this namespace

Classes

- [ArmatureDisplay](#)
- [DragonBonesAsset](#)
- [DragonBonesAtlasAsset](#)

Details

js Module

This module provides some JavaScript utilities. All members can be accessed with "cc.js".

Classes

- [array](#)
- [Pool](#)

Index

Methods

- `isNumber` If a number is created by using 'new Number(10086)', the typeof it will be "object"...
- `isString` Check the obj whether is string or not.
- `addon` Copy all properties not defined in obj from arguments[1...n]
- `mixin` copy all properties from arguments[1...n] to obj
- `extend` Derive the class from the supplied base class.
- `getSuper` Get super class
- `isChildClassOf` Checks whether subclass is child of superclass or equals to superclass
- `clear` Removes all enumerable properties from object
- `getPropertyDescriptor` Get property descriptor in object and all its ancestors
- `value` Define value, just help to call Object.defineProperty.
- ...
- `getset` Define get set accessor, just help to call Object.defineProperty(...)
- `get` Define get accessor, just help to call Object.defineProperty(...)
- `set` Define set accessor, just help to call Object.defineProperty(...)
- `getClassName` Get class name of the object, if object is just a {} (and which class named 'Object'), it will return "".

- `_setClassId` Register the class by specified id, if its classname is not defined, the class name will also be set.
- `setClassName` Register the class by specified name manually
- `unregisterClass` Unregister a class from fireball.
- `_getClassById` Get the registered class by id
- `getClassByName` Get the registered class by name
- `_getClassId` Get class id of the object
- `obsolete` Defines a polyfill field for obsoleted codes.
- `obsoletes` Defines all polyfill fields for obsoleted codes corresponding to the enumerable properties of props.
- `formatStr` A string tool to construct a string with format string.
- `createMap` A simple wrapper of `Object.create(null)` which ensures the return object have no prototype (and thus no inherited members).

Details

Methods

isNumber

Check the obj whether is number or not If a number is created by using 'new Number(10086)', the typeof it will be "object"... Then you can use this function if you care about this case.

| meta | description |
|------------|--|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/js.js:58 |

Parameters

- `obj` Any

isString

Check the obj whether is string or not. If a string is created by using 'new String("blabla")', the typeof it will be "object"... Then you can use this function if you care about this case.

| meta | description |
|---------|-------------------------|
| Returns | Boolean |

| meta | description |
|-------------|--|
| Defined in | cocos2d/core/platform/js.js:70 |

Parameters

- `obj` Any

addon

Copy all properties not defined in obj from arguments[1...n]

| meta | description |
|-------------|--|
| Returns | Object |
| Defined in | cocos2d/core/platform/js.js:82 |

Parameters

- `obj` [Object](#) object to extend its properties
- `sourceObj` [Object](#) source object to copy properties from

Mixin

copy all properties from arguments[1...n] to obj

| meta | description |
|-------------|---|
| Returns | Object |
| Defined in | cocos2d/core/platform/js.js:109 |

Parameters

- `obj` [Object](#)
- `sourceObj` [Object](#)

extend

Derive the class from the supplied base class. Both classes are just native javascript constructors, not created by cc.Class, so usually you will want to inherit using [cc.Class](#) instead.

| meta | description |
|------------|---|
| Returns | Function |
| Defined in | cocos2d/core/platform/js.js:134 |

Parameters

- `cls` [Function](#)
- `base` [Function](#) the baseclass to inherit

getSuper

Get super class

| meta | description |
|------------|---|
| Returns | Function |
| Defined in | cocos2d/core/platform/js.js:168 |

Parameters

- `ctor` [Function](#) the constructor of subclass

isChildClassOf

Checks whether subclass is child of superclass or equals to superclass

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/js.js:180 |

Parameters

- subclass [Function](#)
- superclass [Function](#)

clear

Removes all enumerable properties from object

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/js.js:215 |

Parameters

- obj Any

getPropertyDescriptor

Get property descriptor in object and all its ancestors

| meta | description |
|------------|---|
| Returns | Object |
| Defined in | cocos2d/core/platform/js.js:227 |

Parameters

- obj [Object](#)
- name [String](#)

value

Define value, just help to call Object.defineProperty.
The configurable will be true.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/js.js:245 |

Parameters

- obj [Object](#)
- prop [String](#)
- value Any
- writable [Boolean](#)
- enumerable [Boolean](#)

getset

Define get set accessor, just help to call Object.defineProperty(...)

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/js.js:269 |

Parameters

- obj [Object](#)
- prop [String](#)
- getter [Function](#)
- setter [Function](#)
- enumerable [Boolean](#)

get

Define get accessor, just help to call Object.defineProperty(...)

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/js.js:297 |

Parameters

- obj [Object](#)
- prop [String](#)
- getter [Function](#)
- enumerable [Boolean](#)
- configurable [Boolean](#)

set

Define set accessor, just help to call Object.defineProperty(...)

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/js.js:320 |

Parameters

- `obj` [Object](#)
- `prop` [String](#)
- `setter` [Function](#)
- `enumerable` [Boolean](#)
- `configurable` [Boolean](#)

getClassName

Get class name of the object, if object is just a {} (and which class named 'Object'), it will return "". (modified from [the code from this stackoverflow post](#))

| meta | description |
|------------|---|
| Returns | String |
| Defined in | cocos2d/core/platform/js.js:337 |

Parameters

- `objOrCtor` [Object](#) | [Function](#) instance or constructor

_setClassId

Register the class by specified id, if its classname is not defined, the class name will also be set.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/js.js:415 |

Parameters

- `classId` [String](#)
- `constructor` [Function](#)

setClassName

Register the class by specified name manually

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/js.js:426 |

Parameters

- `className` [String](#)
- `constructor` [Function](#)

unregisterClass

Unregister a class from fireball.

If you dont need a registered class anymore, you should unregister the class so that Fireball will not keep its reference anymore. Please note that its still your responsibility to free other references to the class.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/js.js:443 |

Parameters

- `constructor` [Function](#) the class you will want to unregister, any number of classes can be added

_getClassById

Get the registered class by id

| meta | description |
|------------|---|
| Returns | Function |
| Defined in | cocos2d/core/platform/js.js:466 |

Parameters

- `classId` [String](#)

`getClassName`

Get the registered class by name

| meta | description |
|------------|---|
| Returns | Function |
| Defined in | cocos2d/core/platform/js.js:477 |

Parameters

- `classname` [String](#)

`_getClassId`

Get class id of the object

| meta | description |
|------------|---|
| Returns | String |
| Defined in | cocos2d/core/platform/js.js:487 |

Parameters

- `obj` [Object](#) | [Function](#) instance or constructor
- `allowTempId` [Boolean](#) can return temp id in editor

`obsolete`

Defines a polyfill field for obsoleted codes.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/js.js:554 |

Parameters

- `obj` Any YourObject or YourClass.prototype
- `obsoleted` [String](#) "OldParam" or "YourClass.OldParam"
- `newExpr` [String](#) "NewParam" or "YourClass.NewParam"
- `writable` [Boolean](#)

obsoletes

Defines all polyfill fields for obsoleted codes corresponding to the enumerable properties of props.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/js.js:588 |

Parameters

- `obj` Any YourObject or YourClass.prototype
- `objName` Any "YourObject" or "YourClass"
- `props` [Object](#)
- `writable` [Boolean](#)

formatStr

A string tool to construct a string with format string.

| meta | description |
|------------|---|
| Returns | String |
| Defined in | cocos2d/core/platform/js.js:606 |

Parameters

- `msg` [String](#) | Any A JavaScript string containing zero or more substitution strings (%s).
- `subst` Any JavaScript objects with which to replace substitution strings within msg. This gives you additional control over the format of the output.

Examples

```
cc.js.formatStr("a: %s, b: %s", a, b);
cc.js.formatStr(a, b, c);
```

createMap

A simple wrapper of `Object.create(null)` which ensures the return object have no prototype (and thus no inherited members). So we can skip `hasOwnProperty` calls on property lookups. It is a worthwhile optimization than the `{}` literal when `hasOwnProperty` calls are necessary.

| meta | description |
|------------|---|
| Returns | Object |
| Defined in | cocos2d/core/platform/js.js:655 |

Parameters

- `forceDictMode Boolean` Apply the delete operator to newly created map object. This causes V8 to put the object in "dictionary mode" and disables creation of hidden classes which are very expensive for objects that are constantly changing shape.

sp Module

The global main namespace of Spine, all classes, functions, properties and constants of Spine are defined in this namespace

Classes

- [Skeleton](#)
- [SkeletonData](#)

Enums

- [AccountGender](#)
- [AccountOperate](#)
- [AccountType](#)
- [AdsPos](#)
- [AdsresultCode](#)
- [AdsType](#)
- [AnimationEventType](#)
- [audioEngine.AudioState](#)
- [Button.Transition](#)
- [Camera.ClearFlags](#)
- [CustomresultCode](#)
- [debug.DebugMode](#)
- [DrawBits](#)

- `EditBox.InputFlag`
- `EditBox.InputMode`
- `EditBox.KeyboardReturnType`
- `Graphics.LineCap`
- `Graphics.LineJoin`
- `Label.HorizontalAlign`
- `Label.Overflow`
- `Label.Type`
- `Label.VerticalAlign`
- `Layout.AxisDirection`
- `Layout.HorizontalDirection`
- `Layout.ResizeMode`
- `Layout.Type`
- `Layout.VerticalDirection`
- `LoadingItems.ItemState`
- `macro.BlendFactor`
- `macro.ImageFormat`
- `macro.KEY`
- `macro.TextAlignment`
- `Mask.Type`
- `NetworkType`
- `PageView.Direction`
- `PageView.EventType`
- `PageView.SizeMode`
- `PageViewIndicator.Direction`
- `ParticleSystem.EmitterMode`
- `ParticleSystem.PositionType`
- `PayResultCode`
- `Prefab.OptimizationPolicy`
- `ProgressBar.Mode`
- `PushActionResultCode`
- `RayCastType`
- `RECResultCode`
- `RigidBodyType`
- `Scrollbar.Direction`
- `ScrollView.EventType`
- `ShareResultCode`
- `Slider.Direction`
- `SocialRetCode`
- `Sprite.FillType`
- `Sprite.SizeMode`
- `Sprite.Type`
- `TaskType`
- `Texture2D.Filter`
- `Texture2D.PixelFormat`
- `Texture2D.WrapMode`

- [TiledMap.Orientation](#)
- [ToolBarPlace](#)
- [UserActionResultCode](#)
- [VerticalTextAlignment](#)
- [VideoPlayer.EventType](#)
- [VideoPlayer.ResourceType](#)
- [WebView.EventType](#)
- [Widget.AlignMode](#)
- [WrapMode](#)

sp.spine Module

`sp.spine` is the namespace for official Spine Runtime, which officially implemented and maintained by Spine.

Please refer to the official documentation for its detailed usage: <http://en.esotericsoftware.com/spine-using-runtimes>

Action Class

Defined in: <https://github.com/cocos-creator/engine/blob/4f734a806d1fd7c4073fb064fddc961384fe67af/cocos2d/actions/CAction.js:38>

Module: [cc](#)

Base class cc.Action for action classes.

Index

Properties

- [TAG_INVALID](#) Number Default Action tag.

Methods

- [clone](#) to copy object with deep copy.
- [isDone](#) return true if the action has finished.
- [getTarget](#) get the target.
- [setTarget](#) The action will modify the target properties.
- [getOriginalTarget](#) get the original target.
- [getTag](#) get tag number.
- [setTag](#) set tag number.

Details

Properties

TAG_INVALID

Default Action tag.

| meta | description |
|------------|---|
| Type | Number |
| Defined in | cocos2d/actions/CCAction.js:172 |

Methods

clone

to copy object with deep copy. returns a clone of action.

| meta | description |
|------------|--|
| Returns | Action |
| Defined in | cocos2d/actions/CCAction.js:54 |

isDone

return true if the action has finished.

| meta | description |
|------------|--|
| Returns | Boolean |
| Defined in | cocos2d/actions/CCAction.js:70 |

getTarget

get the target.

| meta | description |
|------------|---|
| Returns | Node |
| Defined in | cocos2d/actions/CCAction.js:102 |

setTarget

The action will modify the target properties.

| meta | description |
|------------|---|
| Defined in | cocos2d/actions/CCAction.js:112 |

Parameters

- target [Node](#)

getOriginalTarget

get the original target.

| meta | description |
|------------|---|
| Returns | Node |
| Defined in | cocos2d/actions/CCAction.js:122 |

getTag

get tag number.

| meta | description |
|------------|---|
| Returns | Number |
| Defined in | cocos2d/actions/CCAction.js:139 |

setTag

set tag number.

| meta | description |
|------------|---|
| Defined in | cocos2d/actions/CCAction.js:149 |

Parameters

- `tag` [Number](#)

ActionInstant Class

Extends [FiniteTimeAction](#)

Module: [cc](#)

Instant actions are immediate actions. They don't have a duration like the ActionInterval actions.

Index

Methods

- [getDuration](#) get duration of the action.
- [setDuration](#) set duration of the action.
- [reverse](#) Returns a reversed action.
- [clone](#) to copy object with deep copy.
- [isDone](#) return true if the action has finished.
- [getTarget](#) get the target.
- [setTarget](#) The action will modify the target properties.
- [getOriginalTarget](#) get the original target.
- [getTag](#) get tag number.
- [setTag](#) set tag number.

Details

Methods

getDuration

get duration of the action. (seconds).

| meta | description |
|------------|---|
| Returns | Number |
| Defined in | cocos2d/actions/CCAction.js:205 |

setDuration

set duration of the action. (seconds).

| meta | description |
|------------|---|
| Defined in | cocos2d/actions/CCAction.js:215 |

Parameters

- duration [Number](#)

reverse

Returns a reversed action.

For example:

- The action will be x coordinates of 0 move to 100.
- The reversed action will be x of 100 move to 0.
- Will be rewritten

| meta | description |
|------------|---|
| Returns | Null |
| Defined in | cocos2d/actions/CCAction.js:225 |

clone

to copy object with deep copy. returns a clone of action.

| meta | description |
|------------|---|
| Returns | FiniteTimeAction |
| Defined in | cocos2d/actions/CCAction.js:241 |

isDone

return true if the action has finished.

| meta | description |
|------------|--|
| Returns | Boolean |
| Defined in | cocos2d/actions/CCAction.js:70 |

getTarget

get the target.

| meta | description |
|------------|---|
| Returns | Node |
| Defined in | cocos2d/actions/CCAction.js:102 |

setTarget

The action will modify the target properties.

| meta | description |
|------------|---|
| Defined in | cocos2d/actions/CCAction.js:112 |

Parameters

- target [Node](#)

getOriginalTarget

get the original target.

| meta | description |
|------------|---|
| Returns | Node |
| Defined in | cocos2d/actions/CCAction.js:122 |

getTag

get tag number.

| meta | description |
|------------|---|
| Returns | Number |
| Defined in | cocos2d/actions/CCAction.js:139 |

setTag

set tag number.

| meta | description |
|------------|---|
| Defined in | cocos2d/actions/CCAction.js:149 |

Parameters

- tag [Number](#)

ActionManager Class

Module: [cc](#)

cc.ActionManager is a class that can manage actions.

Normally you won't need to use this class directly. 99% of the cases you will use the CCNode interface, which uses this class's singleton object. But there are some cases where you might need to use this class.

Examples:

- When you want to run an action where the target is different from a CCNode.
- When you want to pause / resume the actions

Index

Methods

- `addAction` If the target is already present, then the action will be added to the existing target.
- `removeAllActions` Removes all actions from all the targets.
- `removeAllActionsFromTarget` Removes all actions from a certain target.
- `removeAction` Removes an action given an action reference.
- `removeActionByTag` Removes an action given its tag and the target.
- `getActionByTag` Gets an action given its tag an a target.
- `getNumberOfRunningActionsInTarget` Returns the numbers of actions that are running in a certain target.
- `pauseTarget` Pauses the target: all running actions and newly added actions will be paused.
- `resumeTarget` Resumes the target.
- `pauseAllRunningActions` Pauses all running actions, returning a list of targets whose actions were paused.
- `resumeTargets` Resume a set of targets (convenience function to reverse a pauseAllRunningActions or pauseTargets call).
- `pauseTargets` Pause a set of targets.
- `purgeSharedManager` purges the shared action manager.
- `update` The ActionManager update.

Details

Methods

`addAction`

Adds an action with a target.

If the target is already present, then the action will be added to the existing target. If the target is not present, a new instance of this target will be created either paused or not, and the action will be added to the newly created target. When the target is paused, the queued actions won't be 'ticked'.

| meta | description |
|------------|--|
| Defined in | cocos2d/actions/CCActionManager.js:105 |

Parameters

- `action` [Action](#)

- target [Node](#)
- paused [Boolean](#)

removeAllActions

Removes all actions from all the targets.

| meta | description |
|------------|--|
| Defined in | cocos2d/actions/CCActionManager.js:144 |

removeAllActionsFromTarget

Removes all actions from a certain target.

All the actions that belongs to the target will be removed.

| meta | description |
|------------|--|
| Defined in | cocos2d/actions/CCActionManager.js:157 |

Parameters

- target [Node](#)
- forceDelete [Boolean](#)

removeAction

Removes an action given an action reference.

| meta | description |
|------------|--|
| Defined in | cocos2d/actions/CCActionManager.js:178 |

Parameters

- action [Action](#)

removeActionByTag

Removes an action given its tag and the target.

| meta | description |
|-------------|--|
| Defined in | cocos2d/actions/CCActionManager.js:206 |

Parameters

- tag [Number](#)
- target [Node](#)

getActionByTag

Gets an action given its tag an a target.

| meta | description |
|-------------|--|
| Returns | Action Null |
| Defined in | cocos2d/actions/CCActionManager.js:233 |

Parameters

- tag [Number](#)
- target [Node](#)

getNumberOfRunningActionsInTarget

Returns the numbers of actions that are running in a certain target.
Composable actions are counted as 1 action.

Example:

- If you are running 1 Sequence of 7 actions, it will return 1.
- If you are running 7 Sequences of 2 actions, it will return 7.

| meta | description |
|-------------|--|
| Returns | Number |
| Defined in | cocos2d/actions/CCActionManager.js:260 |

Parameters

- target [Node](#)

pauseTarget

Pauses the target: all running actions and newly added actions will be paused.

| meta | description |
|------------|--|
| Defined in | cocos2d/actions/CCActionManager.js:285 |

Parameters

- target [Node](#)

resumeTarget

Resumes the target. All queued actions will be resumed.

| meta | description |
|------------|--|
| Defined in | cocos2d/actions/CCActionManager.js:296 |

Parameters

- target [Node](#)

pauseAllRunningActions

Pauses all running actions, returning a list of targets whose actions were paused.

| meta | description |
|------------|--|
| Returns | Array |
| Defined in | cocos2d/actions/CCActionManager.js:308 |

resumeTargets

Resume a set of targets (convenience function to reverse a pauseAllRunningActions or pauseTargets call).

| meta | description |
|------------|--|
| Defined in | cocos2d/actions/CCActionManager.js:327 |

Parameters

- `targetsToResume` [Array](#)

pauseTargets

Pause a set of targets.

| meta | description |
|------------|--|
| Defined in | cocos2d/actions/CCActionManager.js:343 |

Parameters

- `targetsToPause` [Array](#)

purgeSharedManager

purges the shared action manager. It releases the retained instance.
because it uses this, so it can not be static.

| meta | description |
|------------|--|
| Defined in | cocos2d/actions/CCActionManager.js:359 |

update

The ActionManager update.

| meta | description |
|------------|--|
| Defined in | cocos2d/actions/CCActionManager.js:406 |

Parameters

- `dt` [Number](#) delta time in seconds

AffineTransform Class

Module: [cc](#)

AffineTransform class represent an affine transform matrix. It's composed basically by translation, rotation, scale transformations.

Index

Methods

- [create](#) Create a AffineTransform object with all contents in the matrix.
- [identity](#) Create a identity transformation matrix: ...
- [clone](#) Clone a AffineTransform object from the specified transform.
- [concat](#) Concatenate a transform matrix to another
- [invert](#) Get the invert transform of an AffineTransform object.
- [invert](#) Get an AffineTransform object from a given matrix 4x4.
- [transformVec2](#) Apply the affine transformation on a point.
- [transformSize](#) Apply the affine transformation on a size.
- [transformRect](#) Apply the affine transformation on a rect.
- [transformOBB](#) Apply the affine transformation on a rect, and trans to an Oriented Bounding Box.

Details

Methods

create

Create a AffineTransform object with all contents in the matrix.

| meta | description |
|---------|---------------------------------|
| Returns | AffineTransform |

| meta | description |
|------------|---|
| Defined in | cocos2d/core/utils/affine-transform.js:55 |

Parameters

- a [Number](#)
- b [Number](#)
- c [Number](#)
- d [Number](#)
- tx [Number](#)
- ty [Number](#)

identity

Create a identity transformation matrix:

```
[ 1, 0, 0,
 0, 1, 0 ]
```

| meta | description |
|------------|---|
| Returns | AffineTransform |
| Defined in | cocos2d/core/utils/affine-transform.js:72 |

clone

Clone a AffineTransform object from the specified transform.

| meta | description |
|------------|---|
| Returns | AffineTransform |
| Defined in | cocos2d/core/utils/affine-transform.js:90 |

Parameters

- t [AffineTransform](#)

concat

Concatenate a transform matrix to another The results are reflected in the out affine transform $out = t1 * t2$ This function is memory free, you should create the output affine transform by yourself and manage its memory.

| meta | description |
|------------|--|
| Returns | AffineTransform |
| Defined in | cocos2d/core/utils/affine-transform.js:102 |

Parameters

- `out` [AffineTransform](#) Out object to store the concat result
- `t1` [AffineTransform](#) The first transform object.
- `t2` [AffineTransform](#) The transform object to concatenate.

invert

Get the invert transform of an AffineTransform object. This function is memory free, you should create the output affine transform by yourself and manage its memory.

| meta | description |
|------------|--|
| Returns | AffineTransform |
| Defined in | cocos2d/core/utils/affine-transform.js:129 |

Parameters

- `out` [AffineTransform](#)
- `t` [AffineTransform](#)

invert

Get an AffineTransform object from a given matrix 4x4. This function is memory free, you should create the output affine transform by yourself and manage its memory.

| meta | description |
|------------|--|
| Returns | AffineTransform |
| Defined in | cocos2d/core/utils/affine-transform.js:152 |

Parameters

- `out` [AffineTransform](#)
- `t` [AffineTransform](#)

transformVec2

Apply the affine transformation on a point. This function is memory free, you should create the output Vec2 by yourself and manage its memory.

| meta | description |
|------------|--|
| Returns | Vec2 |
| Defined in | cocos2d/core/utils/affine-transform.js:172 |

Parameters

- `out` [Vec2](#) The output point to store the result
- `point` [Vec2](#) | [Number](#) Point to apply transform or x.
- `transOrY` [AffineTransform](#) | [Number](#) transform matrix or y.
- `t` [AffineTransform](#) transform matrix.

transformSize

Apply the affine transformation on a size. This function is memory free, you should create the output Size by yourself and manage its memory.

| meta | description |
|------------|--|
| Returns | Size |
| Defined in | cocos2d/core/utils/affine-transform.js:199 |

Parameters

- `out` [Size](#) The output point to store the result
- `size` [Size](#)
- `t` [AffineTransform](#)

transformRect

Apply the affine transformation on a rect. This function is memory free, you should create the output Rect by yourself and manage its memory.

| meta | description |
|------------|--|
| Returns | Rect |
| Defined in | cocos2d/core/utils/affine-transform.js:216 |

Parameters

- `out` [Rect](#)
- `rect` [Rect](#)
- `anAffineTransform` [AffineTransform](#)

transformOBB

Apply the affine transformation on a rect, and turns to an Oriented Bounding Box. This function is memory free, you should create the output vectors by yourself and manage their memory.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/utils/affine-transform.js:253 |

Parameters

- `out_b1` [Vec2](#)
- `out_t1` [Vec2](#)
- `out_tr` [Vec2](#)
- `out_br` [Vec2](#)
- `rect` [Rect](#)
- `anAffineTransform` [AffineTransform](#)

AgentManager Class

Module: [anysdk](#)

agent manager of plugin

Index

Methods

- `init` after the packing tool client finishes creating the game.
- `loadAllPlugins` load all plugins, the operation includes SDK's initialization
- `unloadAllPlugins` unload all plugins
- `getUserPlugin` get user system plugin
- `getIAPPPlugins` get IAP system plugins
- `getIAPPPlugin` get IAP system plugin
- `getSocialPlugin` get social system plugin
- `getSharePlugin` get share system plugin
- `getAnalyticsPlugin` get analytics system plugin
- `getAdsPlugin` get ads system plugin
- `getPushPlugin` get push system plugin
- `getRECPlugin` get REC system plugin
- `getCrashPlugin` get crash system plugin
- `getAdTrackingPlugin` get ad track system plugin
- `getCustomPlugin` get custom system plugin
- `getCustomParam` get custom parameter
- `getChannelId` get channel id
- `isAnalyticsEnabled` get status of analytics
- `setIsAnalyticsEnabled` set whether to analytics
- `end` destroy instance
- `getInstance` get instance

Details

Methods

init

AppKey appSecret and privateKey are the only three parameters generated after the packing tool client finishes creating the game. The oauthLoginServer parameter is the API address provided by the game service to login verification

| meta | description |
|------------|--|
| Defined in | extensions/anysdk/jsb_anysdk.js:59 |

Parameters

- `appKey` [String](#)
- `appSecret` [String](#)
- `privateKey` [String](#)
- `oauthLoginServer` [String](#)

loadAllPlugins

load all plugins, the operation includes SDK's initialization

| meta | description |
|------------|--|
| Defined in | extensions/anysdk/jsb_anysdk.js:77 |

Parameters

- `callback` [Function](#)
- `target` [object](#) The object to bind to.

unloadAllPlugins

unload all plugins

| meta | description |
|------------|--|
| Defined in | extensions/anysdk/jsb_anysdk.js:89 |

getUserPlugin

get user system plugin

| meta | description |
|------------|--|
| Returns | <code>anysdk.ProtocolUser</code> |
| Defined in | extensions/anysdk/jsb_anysdk.js:99 |

getIAPPlugins

get IAP system plugins

| meta | description |
|------------|---|
| Returns | anysdk.ProtocolIAP |
| Defined in | extensions/anysdk/jsb_anysdk.js:111 |

getIAPPlugin

get IAP system plugin

| meta | description |
|------------|---|
| Returns | anysdk.ProtocolIAP |
| Defined in | extensions/anysdk/jsb_anysdk.js:123 |

getSocialPlugin

get social system plugin

| meta | description |
|------------|---|
| Returns | anysdk.ProtocolSocial |
| Defined in | extensions/anysdk/jsb_anysdk.js:135 |

getSharePlugin

get share system plugin

| meta | description |
|------------|---|
| Returns | anysdk.ProtocolShare |
| Defined in | extensions/anysdk/jsb_anysdk.js:147 |

getAnalyticsPlugin

get analytics system plugin

| meta | description |
|------------|---|
| Returns | anysdk.ProtocolAnalytics |
| Defined in | extensions/anysdk/jsb_anysdk.js:159 |

getAdsPlugin

get ads system plugin

| meta | description |
|------------|---|
| Returns | anysdk.ProtocolAds |
| Defined in | extensions/anysdk/jsb_anysdk.js:171 |

getPushPlugin

get push system plugin

| meta | description |
|------------|---|
| Returns | anysdk.ProtocolPush |
| Defined in | extensions/anysdk/jsb_anysdk.js:183 |

getRECPlugin

get REC system plugin

| meta | description |
|---------|--------------------|
| Returns | anysdk.ProtocolREC |

| meta | description |
|------------|---|
| Defined in | extensions/anysdk/jsb_anysdk.js:195 |

getCrashPlugin

get crash system plugin

| meta | description |
|------------|---|
| Returns | anysdk.ProtocolCrash |
| Defined in | extensions/anysdk/jsb_anysdk.js:207 |

getAdTrackingPlugin

get ad track system plugin

| meta | description |
|------------|---|
| Returns | anysdk.ProtocolAdTracking |
| Defined in | extensions/anysdk/jsb_anysdk.js:219 |

getCustomPlugin

get custom system plugin

| meta | description |
|------------|---|
| Returns | anysdk.ProtocolCustom |
| Defined in | extensions/anysdk/jsb_anysdk.js:231 |

getCustomParam

get custom parameter

| meta | description |
|------------|---|
| Returns | String |
| Defined in | extensions/anysdk/jsb_anysdk.js:243 |

getChannelId

get channel id

| meta | description |
|------------|---|
| Returns | String |
| Defined in | extensions/anysdk/jsb_anysdk.js:255 |

isAnalyticsEnabled

get status of analytics

| meta | description |
|------------|---|
| Returns | boolean |
| Defined in | extensions/anysdk/jsb_anysdk.js:267 |

setIsAnalyticsEnabled

set whether to analytics

| meta | description |
|------------|---|
| Defined in | extensions/anysdk/jsb_anysdk.js:279 |

Parameters

- enabled [boolean](#)

end

destory instance

| meta | description |
|------------|---|
| Defined in | extensions/anysdk/jsb_anysdk.js:290 |

getInstance

get instance

| meta | description |
|------------|---|
| Returns | anysdk.AgentManager |
| Defined in | extensions/anysdk/jsb_anysdk.js:301 |

Animation Class

Extends [Component](#), [EventTarget](#)(mixin)

Module: [cc](#)

The animation component is used to play back animations.

Animation provide several events to register :

- play : Emit when begin playing animation
- stop : Emit when stop playing animation
- pause : Emit when pause animation
- resume : Emit when resume animation
- lastframe : If animation repeat count is larger than 1, emit when animation play to the last frame
- finished : Emit when finish playing animation

Index

Properties

- [defaultClip](#) AnimationClip Animation will play the default clip when start game.
- [currentClip](#) AnimationClip Current played clip.
- [_clips](#) AnimationClip[] All the clips used in this animation.

- `playOnLoad` Boolean Whether the animation should auto play the default clip when start game.
- `__eventTargets` Array Register all related EventTargets,...
- `node` Node The node this component is attached to.
- `uuid` String The uid for editor.
- `_enabled` Boolean
- `enabled` Boolean indicates whether this component is enabled or not.
- `enabledInHierarchy` Boolean indicates whether this component is enabled and its node is also active in the hierarchy.
- `_isOnLoadCalled` Number Returns a value which used to indicate the onLoad get called or not.
- `_name` String
- `_objFlags` Number
- `name` String The name of the object.
- `isValid` Boolean Indicates whether the object is not yet destroyed.

Methods

- `getClips` Get all the clips used in this animation.
- `play` Plays an animation and stop other animations.
- `playAdditive` Plays an additive animation, it will not stop other animations.
- `stop` Stops an animation named name.
- `pause` Pauses an animation named name.
- `resume` Resumes an animation named name.
- `setCurrentTime` Make an animation named name go to the specified time.
- `getAnimationState` Returns the animation state named name.
- `addClip` Adds a clip to the animation with name newName.
- `removeClip` Remove clip from the animation list.
- `sample` Samples animations at the current state....
- `on` Register animation event callback.
- `off` Unregister animation event callback.
- `update` This is a lifecycle method.
- `lateUpdate` This is a lifecycle method.
- `__preload` __preload is called before every onLoad.
- `onLoad` When attaching to an active node or its node first activated.
- `start` Called before all scripts' update if the Component is enabled the first time.
- `onEnable` This is a lifecycle method.
- `onDisable` This is a lifecycle method.
- `onDestroy` This is a lifecycle method.
- `onFocusInEditor`
- `onLostFocusInEditor`
- `resetInEditor` Called to initialize the component or node's properties when adding the component the first time or when the Reset command is used.
- `addComponent` Adds a component class to the node.
- `getComponent` Returns the component of supplied type if the node has one attached, null if it doesn't....
- `getComponents` Returns all components of supplied Type in the node.
- `getComponentInChildren` Returns the component of supplied type in any of its children using depth first search.
- `getComponentsInChildren` Returns the components of supplied type in self or any of its children using depth first search.

- `_getLocalBounds` If the component's bounding box is different from the node's, you can implement this method to supply `onRestore` for undo/redo operation.
- `schedule` Schedules a custom selector....
- `scheduleOnce` Schedules a callback function that runs only once, with a delay of 0 or larger.
- `unschedule` Unschedules a custom callback function.
- `unscheduleAllCallbacks` unschedule all scheduled callback functions: custom callback functions, and the 'update' callback function....
- `hasEventListener` Checks whether the EventTarget object has any callback registered for a specific type of event.
- `targetOff` Removes all callbacks previously registered with the same target (passed as parameter).
- `once` Register an callback of a specific event type on the EventTarget,....
- `emit` Trigger an event directly with the event name and necessary arguments.
- `dispatchEvent` Send an event with the event object.
- `destroy` Actual object destruction will delayed until before rendering.
- `_destruct` Clear all references in the instance.
- `_onPreDestroy` Called before the object being destroyed.
- `_serialize` The customized serialization for this object.
- `_deserialize` Init this object from the custom serialized data.

Details

Properties

`defaultClip`

Animation will play the default clip when start game.

| meta | description |
|------------|--|
| Type | AnimationClip |
| Defined in | cocos2d/core/components/CCAnimation.js:154 |

`currentClip`

Current played clip.

| meta | description |
|------|-------------------------------|
| Type | AnimationClip |

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCAnimation.js:189 |

_clips

All the clips used in this animation.

| meta | description |
|------------|--|
| Type | AnimationClip[] |
| Defined in | cocos2d/core/components/CCAnimation.js:206 |

playOnLoad

Whether the animation should auto play the default clip when start game.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCAnimation.js:220 |

__eventTargets

Register all related EventTargets, all event callbacks will be removed in `_onPreDestroy`

| meta | description |
|------------|---|
| Type | Array |
| Defined in | cocos2d/core/components/CCComponent.js:61 |

node

The node this component is attached to. A component is always attached to a node.

| meta | description |
|------------|---|
| Type | Node |
| Defined in | cocos2d/core/components/CCComponent.js:75 |

Examples

```
cc.log(comp.node);
```

uuid

The uuid for editor.

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/components/CCComponent.js:106 |

Examples

```
cc.log(comp.uuid);
```

_enabled

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:147 |

enabled

indicates whether this component is enabled or not.

| meta | description |
|------|-------------------------|
| Type | Boolean |

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:154 |

Examples

```
comp.enabled = true;
cc.log(comp.enabled);
```

enabledInHierarchy

indicates whether this component is enabled and its node is also active in the hierarchy.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:185 |

Examples

```
cc.log(comp.enabledInHierarchy);
```

_isOnLoadCalled

Returns a value which used to indicate the onLoad get called or not.

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/components/CCComponent.js:201 |

Examples

```
cc.log(this._isOnLoadCalled > 0);
```

_name

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:76 |

_objFlags

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/platform/CCObject.js:83 |

name

The name of the object.

| meta | description |
|------------|---|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:243 |

Examples

```
obj.name = "New Obj";
```

isValid

Indicates whether the object is not yet destroyed. (It will not be available after being destroyed)

When an object's `destroy` is called, it is actually destroyed after the end of this frame. So `isValid` will return false from the next frame, while `isValid` in the current frame will still be true. If you want to determine whether the current frame has called `destroy`, use `cc.isValid(obj, true)`, but this is often caused by a particular logical requirements, which is not normally required.

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:261 |

Examples

```
var node = new cc.Node();
cc.log(node.isValid);    // true
node.destroy();
cc.log(node.isValid);    // true, still valid in this frame
// after a frame...
cc.log(node.isValid);    // false, destroyed in the end of last frame
```

Methods

getClips

Get all the clips used in this animation.

| meta | description |
|------------|--|
| Returns | AnimationClip[] |
| Defined in | cocos2d/core/components/CCAnimation.js:263 |

play

Plays an animation and stop other animations.

| meta | description |
|------------|--|
| Returns | AnimationState |
| Defined in | cocos2d/core/components/CCAnimation.js:273 |

Parameters

- name [String](#) The name of animation to play. If no name is supplied then the default animation will be played.

- `startTime` [Number](#) play an animation from startTime

Examples

```
var animCtrl = this.node.getComponent(cc.Animation);
animCtrl.play("linear");
```

playAdditive

Plays an additive animation, it will not stop other animations. If there are other animations playing, then will play several animations at the same time.

| meta | description |
|------------|--|
| Returns | AnimationState |
| Defined in | cocos2d/core/components/CCAnimation.js:290 |

Parameters

- `name` [String](#) The name of animation to play. If no name is supplied then the default animation will be played.
- `startTime` [Number](#) play an animation from startTime

Examples

```
// linear_1 and linear_2 at the same time playing.
var animCtrl = this.node.getComponent(cc.Animation);
animCtrl.playAdditive("linear_1");
animCtrl.playAdditive("linear_2");
```

stop

Stops an animation named name. If no name is supplied then stops all playing animations that were started with this Animation.
Stopping an animation also Rewinds it to the Start.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCAnimation.js:339 |

Parameters

- `name` [String](#) The animation to stop, if not supplied then stops all playing animations.

pause

Pauses an animation named name. If no name is supplied then pauses all playing animations that were started with this Animation.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCAnimation.js:361 |

Parameters

- `name` [String](#) The animation to pauses, if not supplied then pauses all playing animations.

resume

Resumes an animation named name. If no name is supplied then resumes all paused animations that were started with this Animation.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCAnimation.js:382 |

Parameters

- `name` [String](#) The animation to resumes, if not supplied then resumes all paused animations.

setCurrentTime

Make an animation named name go to the specified time. If no name is supplied then make all animations go to the specified time.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCAnimation.js:403 |

Parameters

- `time` [Number](#) The time to go to
- `name` [String](#) Specified animation name, if not supplied then make all animations go to the time.

getAnimationState

Returns the animation state named name. If no animation with the specified name, the function will return null.

| meta | description |
|------------|--|
| Returns | AnimationState |
| Defined in | cocos2d/core/components/CCAnimation.js:423 |

Parameters

- name [String](#)

addClip

Adds a clip to the animation with name newName. If a clip with that name already exists it will be replaced with the new clip.

| meta | description |
|------------|--|
| Returns | AnimationState |
| Defined in | cocos2d/core/components/CCAnimation.js:452 |

Parameters

- clip [AnimationClip](#) the clip to add
- newName [String](#)

removeClip

Remove clip from the animation list. This will remove the clip and any animation states based on it. If there are animation states depend on the clip are playing or clip is defaultClip, it will not delete the clip. But if force is true, then will always remove the clip and any animation states based on it. If clip is defaultClip, defaultClip will be reset to null

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCAnimation.js:493 |

Parameters

- `clip` [AnimationClip](#)
- `force` [Boolean](#) If force is true, then will always remove the clip and any animation states based on it.

sample

Samples animations at the current state.

This is useful when you explicitly want to set up some animation state, and sample it once.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCAnimation.js:547 |

Parameters

- `name` [String](#)

on

Register animation event callback. The event arguments will provide the AnimationState which emit the event. When play an animation, will auto register the event callback to the AnimationState, and unregister the event callback from the AnimationState when animation stopped.

| meta | description |
|------------|--|
| Returns | Function |
| Defined in | cocos2d/core/components/CCAnimation.js:570 |

Parameters

- `type` [String](#) A string representing the event type to listen for.
- `callback` [Function](#) The callback that will be invoked when the event is dispatched.

- The callback is ignored if it is a duplicate (the callbacks are unique).
- state [cc.AnimationState](#)
- target [Object](#) The target (this object) to invoke the callback, can be null
- useCapture [Boolean](#) When set to true, the capture argument prevents callback from being invoked when the event's eventPhase attribute value is BUBBLING_PHASE.
- When false, callback will NOT be invoked when event's eventPhase attribute value is CAPTURING_PHASE.
- Either way, callback will be invoked when event's eventPhase attribute value is AT_TARGET.

Examples

```
onPlay: function (type, state) {
    // callback
}

// register event to all animation
animation.on('play', this.onPlay, this);
```

off

Unregister animation event callback.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCAnimation.js:616 |

Parameters

- type [String](#) A string representing the event type being removed.
- callback [Function](#) The callback to remove.
- target [Object](#) The target (this object) to invoke the callback, if it's not given, only callback without target will be removed
- useCapture [Boolean](#) Specifies whether the callback being removed was registered as a capturing callback or not.
 - If not specified, useCapture defaults to false. If a callback was registered twice,
 - one with capture and one without, each must be removed separately. Removal of a capturing callback
 - does not affect a non-capturing version of the same listener, and vice versa.

Examples

```
// unregister event to all animation
animation.off('play', this.onPlay, this);
```

update

Update is called every frame, if the Component is enabled.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:222 |

Parameters

- dt [Number](#) the delta time in seconds it took to complete the last frame

lateUpdate

LateUpdate is called every frame, if the Component is enabled.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:233 |

__preload

__preload is called before every onLoad. It is used to initialize the builtin components internally, to avoid checking whether onLoad is called before every public method calls. This method should be removed if script priority is supported.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:243 |

onLoad

When attaching to an active node or its node first activated. onLoad is always called before any start functions, this allows you to order initialization of scripts.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:254 |

start

Called before all scripts' update if the Component is enabled the first time. Usually used to initialize some logic which need to be called after all components' `onLoad` methods called.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:267 |

onEnable

Called when this component becomes enabled and its node is active.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:280 |

onDisable

Called when this component becomes disabled or its node becomes inactive.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:290 |

onDestroy

Called when this component will be destroyed.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:300 |

onFocusInEditor

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:310 |

onLostFocusInEditor

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:315 |

resetInEditor

Called to initialize the component or node's properties when adding the component the first time or when the Reset command is used. This function is only called in editor.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:320 |

addComponent

Adds a component class to the node. You can also add component to node by passing in the name of the script.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:330 |

Parameters

- typeOrClassName [Function](#) | [String](#) the constructor or the class name of the component to add

Examples

```
var sprite = node.addComponent(cc.Sprite);
var test = node.addComponent("Test");
```

getComponent

Returns the component of supplied type if the node has one attached, null if it doesn't.

You can also get component in the node by passing in the name of the script.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:348 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
// get sprite component.
var sprite = node.getComponent(cc.Sprite);
// get custom test calss.
var test = node.getComponent("Test");
```

getComponents

Returns all components of supplied Type in the node.

| meta | description |
|------------|--|
| Returns | Component[] |
| Defined in | cocos2d/core/components/CCComponent.js:372 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
var sprites = node.getComponents(cc.Sprite);
var tests = node.getComponents("Test");
```

getComponentInChildren

Returns the component of supplied type in any of its children using depth first search.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:390 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
var sprite = node.getComponentInChildren(cc.Sprite);
var Test = node.getComponentInChildren("Test");
```

getComponentsInChildren

Returns the components of supplied type in self or any of its children using depth first search.

| meta | description |
|------------|--|
| Returns | Component[] |
| Defined in | cocos2d/core/components/CCComponent.js:408 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
var sprites = node.getComponentsInChildren(cc.Sprite);
var tests = node.getComponentsInChildren("Test");
```

_getLocalBounds

If the component's bounding box is different from the node's, you can implement this method to supply a custom axis aligned bounding box (AABB), so the editor's scene view can perform hit test properly.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:428 |

Parameters

- `out_rect` [Rect](#) the Rect to receive the bounding box

onRestore

onRestore is called after the user clicks the Reset item in the Inspector's context menu or performs an undo operation on this component.

If the component contains the "internal state", short for "temporary member variables which not included in its CCClass properties", then you may need to implement this function.

The editor will call the getset accessors of your component to record/restore the component's state for undo/redo operation. However, in extreme cases, it may not work well. Then you should implement this function to manually synchronize your component's "internal states" with its public properties. Once you implement this function, all the getset accessors of your component will not be called when the user performs an undo/redo operation. Which means that only the properties with default value will be recorded or restored by editor.

Similarly, the editor may fail to reset your component correctly in extreme cases. Then if you need to support the reset menu, you should manually synchronize your component's "internal states" with its properties in this function. Once you implement this function, all the getset accessors

of your component will not be called during reset operation. Which means that only the properties with default value will be reset by editor.

This function is only called in editor mode.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:441 |

schedule

Schedules a custom selector.

If the selector is already scheduled, then the interval parameter will be updated without scheduling it again.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:535 |

Parameters

- `callback function` The callback function
- `interval Number` Tick interval in seconds. 0 means tick every frame.
- `repeat Number` The selector will be executed (repeat + 1) times, you can use `cc.macro.REPEAT_FOREVER` for tick infinitely.
- `delay Number` The amount of time that the first tick will wait before execution.

Examples

```
var timeCallback = function (dt) {
    cc.log("time: " + dt);
}
this.schedule(timeCallback, 1);
```

scheduleOnce

Schedules a callback function that runs only once, with a delay of 0 or larger.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:572 |

Parameters

- `callback` [function](#) A function wrapped as a selector
- `delay` [Number](#) The amount of time that the first tick will wait before execution.

Examples

```
var timeCallback = function (dt) {  
    cc.log("time: " + dt);  
}  
this.scheduleOnce(timeCallback, 2);
```

unschedule

Unschedules a custom callback function.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:589 |

Parameters

- `callback_fn` [function](#) A function wrapped as a selector

Examples

```
this.unschedule(_callback);
```

unscheduleAllCallbacks

unschedule all scheduled callback functions: custom callback functions, and the 'update' callback function.

Actions are not affected by this method.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:605 |

Examples

```
this.unscheduleAllCallbacks();
```

hasEventListener

Checks whether the EventTarget object has any callback registered for a specific type of event.

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/event/event-target.js:68 |

Parameters

- **type** [String](#) The type of event.

targetOff

Removes all callbacks previously registered with the same target (passed as parameter). This is not for removing all listeners in the current event target, and this is not for removing all listeners the target parameter have registered. It's only for removing all listeners (callback and target couple) registered on the current event target by the target parameter.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/event/event-target.js:151 |

Parameters

- **target** [Object](#) The target to be searched for all related listeners

once

Register an callback of a specific event type on the EventTarget, the callback will remove itself after the first time it is triggered.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/event/event-target.js:164 |

Parameters

- type [String](#) A string representing the event type to listen for.
- callback [Function](#) The callback that will be invoked when the event is dispatched.
- The callback is ignored if it is a duplicate (the callbacks are unique).
- arg1 Any arg1
- arg2 Any arg2
- arg3 Any arg3
- arg4 Any arg4
- arg5 Any arg5
- target [Object](#) The target (this object) to invoke the callback, can be null

Examples

```
eventTarget.once('fire', function (event) {  
    cc.log("this is the callback and will be invoked only once");  
}, node);
```

emit

Trigger an event directly with the event name and necessary arguments.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/event/event-target.js:201 |

Parameters

- type [String](#) event type
- arg1 Any First argument
- arg2 Any Second argument
- arg3 Any Third argument
- arg4 Any Fourth argument
- arg5 Any Fifth argument

Examples

```
eventTarget.emit('fire', event);  
eventTarget.emit('fire', message, emitter);
```

dispatchEvent

Send an event with the event object.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/event/event-target.js:221 |

Parameters

- event [Event](#)

destroy

Destroy this Object, and release all its own references to other objects.
 Actual object destruction will delayed until before rendering. From the next frame, this object is not usable any more. You can use cc.isValid(obj) to check whether the object is destroyed before accessing it.

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:296 |

Examples

```
obj.destroy();
```

_destruct

Clear all references in the instance.

NOTE: this method will not clear the getter or setter functions which defined in the instance of CCObject. You can override the _destruct method if you need, for example:

```
_destruct: function () { for (var key in this) { if (this.hasOwnProperty(key)) { switch (typeof this[key]) { case 'string': this[key] = ""; break; case 'object': case 'function': this[key] = null; break; } } }
```

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:430 |

_onPreDestroy

Called before the object being destroyed.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:463 |

_serialize

The customized serialization for this object. (Editor Only)

| meta | description |
|------------|---|
| Returns | object |
| Defined in | cocos2d/core/platform/CCObject.js:488 |

Parameters

- `exporting Boolean`

_deserialize

Init this object from the custom serialized data.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:498 |

Parameters

- `data Object` the serialized json data
- `ctx _Deserializer`

Animation.EventType Class

Module: [cc](#)

The event type supported by Animation

Index

Properties

- **PLAY** String Emit when begin playing animation
- **STOP** String Emit when stop playing animation
- **PAUSE** String Emit when pause animation
- **RESUME** String Emit when resume animation
- **LASTFRAME** String If animation repeat count is larger than 1, emit when animation play to the last frame
- **FINISHED** String Emit when finish playing animation

Details

Properties

PLAY

Emit when begin playing animation

| meta | description |
|------------|---|
| Type | String |
| Defined in | cocos2d/core/components/CCAnimation.js:51 |

STOP

Emit when stop playing animation

| meta | description |
|------------|---|
| Type | String |
| Defined in | cocos2d/core/components/CCAnimation.js:58 |

PAUSE

Emit when pause animation

| meta | description |
|------|------------------------|
| Type | String |

| meta | description |
|-------------|---|
| Defined in | cocos2d/core/components/CCAnimation.js:65 |

RESUME

Emit when resume animation

| meta | description |
|-------------|---|
| Type | String |
| Defined in | cocos2d/core/components/CCAnimation.js:72 |

LASTFRAME

If animation repeat count is larger than 1, emit when animation play to the last frame

| meta | description |
|-------------|---|
| Type | String |
| Defined in | cocos2d/core/components/CCAnimation.js:79 |

FINISHED

Emit when finish playing animation

| meta | description |
|-------------|---|
| Type | String |
| Defined in | cocos2d/core/components/CCAnimation.js:86 |

AnimationClip Class

Extends [Asset](#)

Module: [cc](#)

Class for animation data handling.

Index

Properties

- `duration` Number Duration of this animation.
- `sample` Number FrameRate of this animation.
- `speed` Number Speed of this animation.
- `wrapMode` WrapMode WrapMode of this animation.
- `curveData` Object Curve data.
- `events` Object[] Event data.
- `loaded` Boolean Whether the asset is loaded or not
- `nativeUrl` String Returns the url of this asset's native object, if none it will returns an empty string.
- `_native` String Serializable url for native asset.
- `_nativeAsset` Object The underlying native asset of this asset if one is available.
- `_uuid` String
- `_name` String
- `_objFlags` Number
- `name` String The name of the object.
- `isValid` Boolean Indicates whether the object is not yet destroyed.

Methods

- `createWithSpriteFrames` Crate clip with a set of sprite frames
- `toString` Returns the asset's url.
- `serialize` 应 AssetDB 要求提供这个方法
- `createNode` Create a new node using this asset in the scene....
- `_setRawAsset` Set native file name for this asset.
- `destroy` Actual object destruction will delayed until before rendering.
- `_destruct` Clear all references in the instance.
- `_onPreDestroy` Called before the object being destroyed.
- `_serialize` The customized serialization for this object.
- `_deserialize` Init this object from the custom serialized data.

Details

Properties

duration

Duration of this animation.

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/animation/animation-clip.js:46 |

sample

FrameRate of this animation.

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/animation/animation-clip.js:56 |

speed

Speed of this animation.

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/animation/animation-clip.js:66 |

wrapMode

WrapMode of this animation.

| meta | description |
|------------|--|
| Type | WrapMode |
| Defined in | cocos2d/animation/animation-clip.js:76 |

curveData

Curve data.

| meta | description |
|------------|--|
| Type | Object |
| Defined in | cocos2d/animation/animation-clip.js:86 |

Examples

```
{
    // 根节点不用查找路径
    // root properties
    props: {
        x: [
            { frame: 0, value: 0, curve: [0,0.5,0.5,1] },
            { frame: 1, value: 200, curve: null }
        ]
    },
    comps: {
        // component
        'comp-1': {
            // component properties
            'prop-1': [
                { frame: 0, value: 10, curve: [0,0.5,0.5,1] },
                { frame: 1, value: 20, curve: null }
            ]
        }
    },
    paths: {
        // key 为节点到 root 的路径名，通过 cc.find 找到
        'foo/bar': {
            // node properties
            props: {
                x: [
                    { frame: 0, value: 0, curve: [0,0.5,0.5,1] },
                    { frame: 1, value: 200, curve: null }
                ]
            }
        }
    }
}
```

```

        },
        comps: {
            // component
            'comp-1': {
                // component property
                'prop-1': [
                    { frame: 0, value: 10, curve: [0,0.5,0,
                        { frame: 1, value: 20, curve: null }
                    ]
                }
            }
        },
        'hello': {
            props: {
                position: [
                    {
                        frame: 0,
                        value: [0,0],
                        motionPath: [
                            [320, 240, 0, 240, 640, 240],
                            [640, 0, 400, 0, 1000, 0]
                        ]
                    },
                    { frame: 5, value: [640, 480] }
                ]
            }
        }
    }
}

```

events

Event data.

| meta | description |
|------------|--|
| Type | Object[] |
| Defined in | cocos2d/animation/animation-clip.js:98 |

Examples

```

// frame      : The exactly time in second.
// func       : Callback function name
// params     : Callback parameters
[
    { frame: 0, func: 'onAnimationEvent1', params:[ 'param-1', 'param-2' ] },
    { frame: 2, func: 'onAnimationEvent3', params:[ 'param-1', 'param-2' ] },
    { frame: 3, func: 'onAnimationEvent2', params:[ 'param-1' ] },

```

```

    // The second event at frame 3
    { frame: 3, func: 'onAnimationEvent4', params:[ 'param-1' ] },
    { frame: 4, func: 'onAnimationEvent4', params:[ 'param-1' ] }
]

```

loaded

Whether the asset is loaded or not

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/assets/CCAsset.js:57 |

nativeUrl

Returns the url of this asset's native object, if none it will returns an empty string.

| meta | description |
|------------|---|
| Type | String |
| Defined in | cocos2d/core/assets/CCAsset.js:70 |

_native

Serializable url for native asset.

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/assets/CCAsset.js:108 |

_nativeAsset

The underlying native asset of this asset if one is available. This property can be used to access additional details or functionality related to the asset. This property will be initialized by the loader if `_native` is available.

| meta | description |
|------------|--|
| Type | Object |
| Defined in | cocos2d/core/assets/CCAsset.js:116 |

_uuid

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/assets/CCRawAsset.js:46 |

_name

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:76 |

_objFlags

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/platform/CCObject.js:83 |

name

The name of the object.

| meta | description |
|------------|---|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:243 |

Examples

```
obj.name = "New Obj";
```

isValid

Indicates whether the object is not yet destroyed. (It will not be available after being destroyed)

When an object's `destroy` is called, it is actually destroyed after the end of this frame. So `isValid` will return false from the next frame, while `isValid` in the current frame will still be true. If you want to determine whether the current frame has called `destroy`, use `cc.isValid(obj, true)`, but this is often caused by a particular logical requirements, which is not normally required.

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:261 |

Examples

```
var node = new cc.Node();
cc.log(node.isValid);    // true
node.destroy();
cc.log(node.isValid);    // true, still valid in this frame
// after a frame...
cc.log(node.isValid);    // false, destroyed in the end of last frame
```

Methods

createWithSpriteFrames

Create clip with a set of sprite frames

| meta | description |
|------------|---|
| Returns | AnimationClip |
| Defined in | cocos2d/animation/animation-clip.js:113 |

Parameters

- `spriteFrames` [SpriteFrame](#)
- `sample` [Number](#)

Examples

```
var clip = cc.AnimationClip.createWithSpriteFrames(spriteFrames, 10);
```

toString

Returns the asset's url.

The Asset object overrides the `toString()` method of the `Object` object. For Asset objects, the `toString()` method returns a string representation of the object. JavaScript calls the `toString()` method automatically when an asset is to be represented as a text value or when a texture is referred to in a string concatenation.

| meta | description |
|------------|--|
| Returns | String |
| Defined in | cocos2d/core/assets/CCAsset.js:165 |

serialize

应 AssetDB 要求提供这个方法

| meta | description |
|------------|--|
| Returns | String |
| Defined in | cocos2d/core/assets/CCAsset.js:179 |

createNode

Create a new node using this asset in the scene.

If this type of asset don't have its corresponding node type, this method should be null.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/assets/CCAsset.js:190 |

Parameters

- callback [Function](#)
- error [String](#) null or the error info
- node [Object](#) the created node or null

_setRawAsset

Set native file name for this asset.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/assets/CCAsset.js:205 |

Parameters

- filename [String](#)
- inLibrary [Boolean](#)

destroy

Destroy this Object, and release all its own references to other objects.

Actual object destruction will be delayed until before rendering. From the next frame, this object is not usable any more. You can use cc.isValid(obj) to check whether the object is destroyed before accessing it.

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:296 |

Examples

```
obj.destroy();
```

_destruct

Clear all references in the instance.

NOTE: this method will not clear the getter or setter functions which defined in the instance of CCObject. You can override the _destruct method if you need, for example:

```
_destruct: function () { for (var key in this) { if (this.hasOwnProperty(key)) { switch (typeof this[key]) { case 'string': this[key] = ""; break; case 'object': case 'function': this[key] = null; break; } } }
```

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:430 |

_onPreDestroy

Called before the object being destroyed.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:463 |

_serialize

The customized serialization for this object. (Editor Only)

| meta | description |
|------------|---|
| Returns | object |
| Defined in | cocos2d/core/platform/CCObject.js:488 |

Parameters

- `exporting Boolean`

_deserialize

Init this object from the custom serialized data.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:498 |

Parameters

- `data Object` the serialized json data
- `ctx _Deserializer`

AnimationState Class

Extends [Playable](#)

Module: [cc](#)

The AnimationState gives full control over animation playback process. In most cases the Animation Component is sufficient and easier to use. Use the AnimationState if you need full control.

Index

Properties

- `animator` AnimationAnimator
- `curves` Object[] The curves list.
- `delay` Number The start delay which represents the number of seconds from an animation's start time to the start of
- `repeatCount` Number The animation's iteration count property.
- `duration` Number The iteration duration of this animation in seconds.
- `speed` Number The animation's playback speed.
- `wrapMode` WrapMode Wrapping mode of the playing animation.
- `time` Number The current time of this animation in seconds.
- `clip` AnimationClip The clip that is being played by this animation state.
- `name` String The name of the playing animation.
- `isPlaying` Boolean Is playing or paused in play mode?
- `isPaused` Boolean Is currently paused? This can be true even if in edit mode(isPlaying == false).

Methods

- `constructor`
- `onPlay`
- `onPause`
- `onResume`
- `onStop`
- `onError`

- [play](#) Play this animation.
- [stop](#) Stop this animation.
- [pause](#) Pause this animation.
- [resume](#) Resume this animation.
- [step](#) Perform a single frame step.

Details

Properties

animator

| meta | description |
|------------|---|
| Type | AnimationAnimator |
| Defined in | cocos2d/animation/animation-state.js:73 |

curves

The curves list.

| meta | description |
|------------|---|
| Type | Object[] |
| Defined in | cocos2d/animation/animation-state.js:79 |

delay

The start delay which represents the number of seconds from an animation's start time to the start of the active interval.

| meta | description |
|------------|---|
| Type | Number |
| Defined in | cocos2d/animation/animation-state.js:89 |

repeatCount

The animation's iteration count property.

A real number greater than or equal to zero (including positive infinity) representing the number of times to repeat the animation node.

Values less than zero and NaN values are treated as the value 1.0 for the purpose of timing model calculations.

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/animation/animation-state.js:100 |

duration

The iteration duration of this animation in seconds. (length)

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/animation/animation-state.js:117 |

speed

The animation's playback speed. 1 is normal playback speed.

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/animation/animation-state.js:127 |

wrapMode

Wrapping mode of the playing animation. Notice : dynamic change wrapMode will reset time and repeatCount property

| meta | description |
|------------|--|
| Type | WrapMode |
| Defined in | cocos2d/animation/animation-state.js:136 |

time

The current time of this animation in seconds.

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/animation/animation-state.js:150 |

clip

The clip that is being played by this animation state.

| meta | description |
|------------|--|
| Type | AnimationClip |
| Defined in | cocos2d/animation/animation-state.js:441 |

name

The name of the playing animation.

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/animation/animation-state.js:452 |

isPlaying

Is playing or paused in play mode?

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/animation/playable.js:44 |

isPaused

Is currently paused? This can be true even if in edit mode(isPlaying == false).

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/animation/playable.js:56 |

Methods

constructor

| meta | description |
|------------|---|
| Defined in | cocos2d/animation/animation-state.js:50 |

Parameters

- `clip` [AnimationClip](#)
- `name` [String](#)

onPlay

| meta | description |
|------------|--|
| Defined in | cocos2d/animation/playable.js:71 |

onPause

| meta | description |
|------------|--|
| Defined in | cocos2d/animation/playable.js:76 |

onResume

| meta | description |
|------------|--|
| Defined in | cocos2d/animation/playable.js:81 |

onStop

| meta | description |
|------------|--|
| Defined in | cocos2d/animation/playable.js:86 |

onError

| meta | description |
|------------|--|
| Defined in | cocos2d/animation/playable.js:91 |

Parameters

- `errorCode` [string](#)

play

Play this animation.

| meta | description |
|------------|---|
| Defined in | cocos2d/animation/playable.js:100 |

stop

Stop this animation.

| meta | description |
|------------|---|
| Defined in | cocos2d/animation/playable.js:121 |

pause

Pause this animation.

| meta | description |
|------------|---|
| Defined in | cocos2d/animation/playable.js:136 |

resume

Resume this animation.

| meta | description |
|------------|---|
| Defined in | cocos2d/animation/playable.js:148 |

step

Perform a single frame step.

| meta | description |
|------------|---|
| Defined in | cocos2d/animation/playable.js:160 |

ArmatureDisplay Class

Extends [Component](#)
Module: [dragonBones](#)

The Armature Display of DragonBones

(Armature Display has a reference to a DragonBonesAsset and stores the state for ArmatureDisplay instance, which consists of the current pose's bone SRT, slot colors, and which slot attachments are visible.

Multiple Armature Display can use the same DragonBonesAsset which includes all animations, skins, and attachments.)

Index

Properties

- `dragonAsset` DragonBonesAsset The DragonBones data contains the armatures information (bind pose bones, slots, draw order,...)
- `dragonAtlasAsset` DragonBonesAtlasAsset The atlas asset for the DragonBones.
- `armatureName` String The name of current armature.
- `animationName` String The name of current playing animation.
- `_defaultArmatureIndex` Number
- `timeScale` Number The time scale of this armature.
- `playTimes` Number The play times of the default animation.
- `debugBones` Boolean Indicates whether open debug bones.
- `__eventTargets` Array Register all related EventTargets,...
- `node` Node The node this component is attached to.
- `uuid` String The uuid for editor.
- `_enabled` Boolean
- `enabled` Boolean indicates whether this component is enabled or not.
- `enabledInHierarchy` Boolean indicates whether this component is enabled and its node is also active in the hierarchy.
- `_isOnLoadCalled` Number Returns a value which used to indicate the onLoad get called or not.
- `_name` String
- `_objFlags` Number
- `name` String The name of the object.
- `isValid` Boolean Indicates whether the object is not yet destroyed.

Methods

- `playAnimation` Play the specified animation.
- `getArmatureNames` Get the all armature names in the DragonBones Data.
- `getAnimationNames` Get the all animation names of specified armature.
- `addEvent Listener` Add event listener for the DragonBones Event.
- `removeEvent Listener` Remove the event listener for the DragonBones Event.
- `buildArmature` Build the armature for specified name.
- `armature` Get the current armature object of the ArmatureDisplay.
- `update` This is a lifecycle method.
- `lateUpdate` This is a lifecycle method.

- `__preload` `__preload` is called before every `onLoad`.
- `onLoad` When attaching to an active node or its node first activated.
- `start` Called before all scripts' update if the Component is enabled the first time.
- `onEnable` This is a lifecycle method.
- `onDisable` This is a lifecycle method.
- `onDestroy` This is a lifecycle method.
- `onFocusInEditor`
- `onLostFocusInEditor`
- `resetInEditor` Called to initialize the component or node's properties when adding the component the first time or when the Reset command is used.
- `addComponent` Adds a component class to the node.
- `getComponent` Returns the component of supplied type if the node has one attached, null if it doesn't....
- `getComponents` Returns all components of supplied Type in the node.
- `getComponentInChildren` Returns the component of supplied type in any of its children using depth first search.
- `getComponentsInChildren` Returns the components of supplied type in self or any of its children using depth first search.
- `_getLocalBounds` If the component's bounding box is different from the node's, you can implement this method to supply
- `onRestore` for undo/redo operation.
- `schedule` Schedules a custom selector....
- `scheduleOnce` Schedules a callback function that runs only once, with a delay of 0 or larger.
- `unschedule` Unschedules a custom callback function.
- `unscheduleAllCallbacks` unschedule all scheduled callback functions: custom callback functions, and the 'update' callback function....
- `destroy` Actual object destruction will delayed until before rendering.
- `_destruct` Clear all references in the instance.
- `_onPreDestroy` Called before the object being destroyed.
- `_serialize` The customized serialization for this object.
- `_deserialize` Init this object from the custom serialized data.

Details

Properties

dragonAsset

The DragonBones data contains the armatures information (bind pose bones, slots, draw order, attachments, skins, etc) and animations but does not hold any state. Multiple ArmatureDisplay can share the same DragonBones data.

| meta | description |
|------|----------------------------------|
| Type | DragonBonesAsset |

| meta | description |
|------------|--|
| Defined in | extensions/dragonbones/ArmatureDisplay.js:86 |

dragonAtlasAsset

The atlas asset for the DragonBones.

| meta | description |
|------------|---|
| Type | DragonBonesAtlasAsset |
| Defined in | extensions/dragonbones/ArmatureDisplay.js:112 |

armatureName

The name of current armature.

| meta | description |
|------------|---|
| Type | String |
| Defined in | extensions/dragonbones/ArmatureDisplay.js:132 |

animationName

The name of current playing animation.

| meta | description |
|------------|---|
| Type | String |
| Defined in | extensions/dragonbones/ArmatureDisplay.js:168 |

_defaultArmatureIndex

| meta | description |
|------------|---|
| Type | Number |
| Defined in | extensions/dragonbones/ArmatureDisplay.js:183 |

timeScale

The time scale of this armature.

| meta | description |
|------------|---|
| Type | Number |
| Defined in | extensions/dragonbones/ArmatureDisplay.js:249 |

playTimes

The play times of the default animation. -1 means using the value of config file; 0 means repeat for ever 0 means repeat times

| meta | description |
|------------|---|
| Type | Number |
| Defined in | extensions/dragonbones/ArmatureDisplay.js:263 |

debugBones

Indicates whether open debug bones.

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | extensions/dragonbones/ArmatureDisplay.js:280 |

eventTargets

Register all related EventTargets, all event callbacks will be removed in _onPreDestroy

| meta | description |
|------------|---|
| Type | Array |
| Defined in | cocos2d/core/components/CCComponent.js:61 |

node

The node this component is attached to. A component is always attached to a node.

| meta | description |
|------------|---|
| Type | Node |
| Defined in | cocos2d/core/components/CCComponent.js:75 |

Examples

```
cc.log(comp.node);
```

uuid

The uuid for editor.

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/components/CCComponent.js:106 |

Examples

```
cc.log(comp.uuid);
```

_enabled

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:147 |

enabled

indicates whether this component is enabled or not.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:154 |

Examples

```
comp.enabled = true;  
cc.log(comp.enabled);
```

enabledInHierarchy

indicates whether this component is enabled and its node is also active in the hierarchy.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:185 |

Examples

```
cc.log(comp.enabledInHierarchy);
```

_isOnLoadCalled

Returns a value which used to indicate the onLoad get called or not.

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/components/CCComponent.js:201 |

Examples

```
cc.log(this._isOnLoadCalled > 0);
```

_name

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:76 |

_objFlags

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/platform/CCObject.js:83 |

name

The name of the object.

| meta | description |
|------------|---|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:243 |

Examples

```
obj.name = "New Obj";
```

isValid

Indicates whether the object is not yet destroyed. (It will not be available after being destroyed)

When an object's `destroy` is called, it is actually destroyed after the end of this frame. So `isValid` will return false from the next frame, while `isValid` in the current frame will still be true. If you want to determine whether the current frame has called `destroy`, use `cc.isValid(obj, true)`, but this is often caused by a particular logical requirements, which is not normally required.

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:261 |

Examples

```
var node = new cc.Node();
cc.log(node.isValid);    // true
node.destroy();
cc.log(node.isValid);    // true, still valid in this frame
// after a frame...
cc.log(node.isValid);    // false, destroyed in the end of last frame
```

Methods

playAnimation

Play the specified animation. Parameter `animName` specify the animation name. Parameter `playTimes` specify the repeat times of the animation. -1 means use the value of the config file. 0 means play the animation for ever.

0 means repeat times.

| meta | description |
|------------|---|
| Returns | <code>dragonBones.AnimationState</code> |
| Defined in | extensions/dragonbones/ArmatureDisplay.js:431 |

Parameters

- `animName` [String](#)
- `playTimes` [Number](#)

getArmatureNames

Get the all armature names in the DragonBones Data.

| meta | description |
|------------|---|
| Returns | Array |
| Defined in | extensions/dragonbones/ArmatureDisplay.js:461 |

getAnimationNames

Get the all animation names of specified armature.

| meta | description |
|------------|---|
| Returns | Array |
| Defined in | extensions/dragonbones/ArmatureDisplay.js:474 |

Parameters

- `armatureName` [String](#)

addEventListener

Add event listener for the DragonBones Event.

| meta | description |
|------------|---|
| Defined in | extensions/dragonbones/ArmatureDisplay.js:499 |

Parameters

- `type` [String](#) A string representing the event type to listen for.
- `listener` [Function](#) The callback that will be invoked when the event is dispatched.

- event [Event](#) event
- target [Object](#) The target (this object) to invoke the callback, can be null

removeEventListener

Remove the event listener for the DragonBones Event.

| meta | description |
|------------|---|
| Defined in | extensions/dragonbones/ArmatureDisplay.js:514 |

Parameters

- type [String](#) A string representing the event type to listen for.
- listener [Function](#)
- target [Object](#)

buildArmature

Build the armature for specified name.

| meta | description |
|------------|---|
| Returns | dragonBones.ArmatureDisplay |
| Defined in | extensions/dragonbones/ArmatureDisplay.js:528 |

Parameters

- armatureName [String](#)
- node [Node](#)

armature

Get the current armature object of the ArmatureDisplay.

| meta | description |
|------------|---|
| Returns | Object |
| Defined in | extensions/dragonbones/ArmatureDisplay.js:542 |

update

Update is called every frame, if the Component is enabled.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:222 |

Parameters

- dt [Number](#) the delta time in seconds it took to complete the last frame

lateUpdate

LateUpdate is called every frame, if the Component is enabled.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:233 |

__preload

__preload is called before every onLoad. It is used to initialize the builtin components internally, to avoid checking whether onLoad is called before every public method calls. This method should be removed if script priority is supported.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:243 |

onLoad

When attaching to an active node or its node first activated. onLoad is always called before any start functions, this allows you to order initialization of scripts.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:254 |

start

Called before all scripts' update if the Component is enabled the first time. Usually used to initialize some logic which need to be called after all components' `onLoad` methods called.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:267 |

onEnable

Called when this component becomes enabled and its node is active.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:280 |

onDisable

Called when this component becomes disabled or its node becomes inactive.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:290 |

onDestroy

Called when this component will be destroyed.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:300 |

onFocusInEditor

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:310 |

onLostFocusInEditor

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:315 |

resetInEditor

Called to initialize the component or node's properties when adding the component the first time or when the Reset command is used. This function is only called in editor.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:320 |

addComponent

Adds a component class to the node. You can also add component to node by passing in the name of the script.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:330 |

Parameters

- typeOrClassName [Function](#) | [String](#) the constructor or the class name of the component to add

Examples

```
var sprite = node.addComponent(cc.Sprite);
var test = node.addComponent("Test");
```

getComponent

Returns the component of supplied type if the node has one attached, null if it doesn't.

You can also get component in the node by passing in the name of the script.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:348 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
// get sprite component.
var sprite = node.getComponent(cc.Sprite);
// get custom test calss.
var test = node.getComponent("Test");
```

getComponents

Returns all components of supplied Type in the node.

| meta | description |
|------------|--|
| Returns | Component[] |
| Defined in | cocos2d/core/components/CCComponent.js:372 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
var sprites = node.getComponents(cc.Sprite);
var tests = node.getComponents("Test");
```

getComponentInChildren

Returns the component of supplied type in any of its children using depth first search.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:390 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
var sprite = node.getComponentInChildren(cc.Sprite);
var Test = node.getComponentInChildren("Test");
```

getComponentsInChildren

Returns the components of supplied type in self or any of its children using depth first search.

| meta | description |
|------------|--|
| Returns | Component[] |
| Defined in | cocos2d/core/components/CCComponent.js:408 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
var sprites = node.getComponentsInChildren(cc.Sprite);
var tests = node.getComponentsInChildren("Test");
```

_getLocalBounds

If the component's bounding box is different from the node's, you can implement this method to supply a custom axis aligned bounding box (AABB), so the editor's scene view can perform hit test properly.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:428 |

Parameters

- `out_rect Rect` the Rect to receive the bounding box

onRestore

onRestore is called after the user clicks the Reset item in the Inspector's context menu or performs an undo operation on this component.

If the component contains the "internal state", short for "temporary member variables which not included in its CCClass properties", then you may need to implement this function.

The editor will call the getset accessors of your component to record/restore the component's state for undo/redo operation. However, in extreme cases, it may not work well. Then you should implement this function to manually synchronize your component's "internal states" with its public properties. Once you implement this function, all the getset accessors of your component will not be called when the user performs an undo/redo operation. Which means that only the properties with default value will be recorded or restored by editor.

Similarly, the editor may fail to reset your component correctly in extreme cases. Then if you need to support the reset menu, you should manually synchronize your component's "internal states" with its properties in this function. Once you implement this function, all the getset accessors

of your component will not be called during reset operation. Which means that only the properties with default value will be reset by editor.

This function is only called in editor mode.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:441 |

schedule

Schedules a custom selector.

If the selector is already scheduled, then the interval parameter will be updated without scheduling it again.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:535 |

Parameters

- `callback function` The callback function
- `interval Number` Tick interval in seconds. 0 means tick every frame.
- `repeat Number` The selector will be executed (repeat + 1) times, you can use `cc.macro.REPEAT_FOREVER` for tick infinitely.
- `delay Number` The amount of time that the first tick will wait before execution.

Examples

```
var timeCallback = function (dt) {
    cc.log("time: " + dt);
}
this.schedule(timeCallback, 1);
```

scheduleOnce

Schedules a callback function that runs only once, with a delay of 0 or larger.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:572 |

Parameters

- `callback` [function](#) A function wrapped as a selector
- `delay` [Number](#) The amount of time that the first tick will wait before execution.

Examples

```
var timeCallback = function (dt) {  
    cc.log("time: " + dt);  
}  
this.scheduleOnce(timeCallback, 2);
```

unschedule

Unschedules a custom callback function.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:589 |

Parameters

- `callback_fn` [function](#) A function wrapped as a selector

Examples

```
this.unschedule(_callback);
```

unscheduleAllCallbacks

unschedule all scheduled callback functions: custom callback functions, and the 'update' callback function.

Actions are not affected by this method.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:605 |

Examples

```
this.unscheduleAllCallbacks();
```

destroy

Destroy this Object, and release all its own references to other objects.

Actual object destruction will be delayed until before rendering. From the next frame, this object is not usable any more. You can use cc.isValid(obj) to check whether the object is destroyed before accessing it.

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:296 |

Examples

```
obj.destroy();
```

_destruct

Clear all references in the instance.

NOTE: this method will not clear the getter or setter functions which defined in the instance of CCObject. You can override the _destruct method if you need, for example:

```
_destruct: function () { for (var key in this) { if (this.hasOwnProperty(key)) { switch (typeof this[key]) { case 'string': this[key] = ''; break; case 'object': case 'function': this[key] = null; break; } } }}
```

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:430 |

_onPreDestroy

Called before the object being destroyed.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:463 |

_serialize

The customized serialization for this object. (Editor Only)

| meta | description |
|------------|---|
| Returns | object |
| Defined in | cocos2d/core/platform/CCObject.js:488 |

Parameters

- `exporting Boolean`

`_deserialize`

Init this object from the custom serialized data.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:498 |

Parameters

- `data Object` the serialized json data
- `ctx _Deserializer`

array Class

Module: [js](#) Parent Module: [js](#)

Index

Methods

- `removeAt` Removes the array item at the specified index.
- `fastRemoveAt` Removes the array item at the specified index.
- `remove` Removes the first occurrence of a specific object from the array.
- `fastRemove` Removes the first occurrence of a specific object from the array.
- `verifyType` Verify array's Type
- `removeArray` Removes from array all values in minusArr.
- `appendObjectsAt` Inserts some objects at index
- `indexOf` Exact same function as Array.prototype.indexOf.
- ...
- `contains` Determines whether the array contains a specific value.
- `copy` Copy an array's item to a new array (its performance is better than Array.slice)

Details

Methods

removeAt

Removes the array item at the specified index.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/js.js:683 |

Parameters

- `array` Any
- `index` [Number](#)

fastRemoveAt

Removes the array item at the specified index. It's faster but the order of the array will be changed.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/js.js:693 |

Parameters

- `array` Any
- `index` [Number](#)

remove

Removes the first occurrence of a specific object from the array.

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/js.js:709 |

Parameters

- `array Any`
- `value Any`

fastRemove

Removes the first occurrence of a specific object from the array. It's faster but the order of the array will be changed.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/js.js:727 |

Parameters

- `array Any`
- `value Number`

verifyType

Verify array's Type

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/js.js:742 |

Parameters

- `array array`
- `type Function`

removeArray

Removes from array all values in minusArr. For each Value in minusArr, the first matching instance in array will be removed.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/js.js:761 |

Parameters

- `array` [Array](#) Source Array
- `minusArr` [Array](#) minus Array

appendObjectsAt

Inserts some objects at index

| meta | description |
|------------|---|
| Returns | Array |
| Defined in | cocos2d/core/platform/js.js:773 |

Parameters

- `array` [Array](#)
- `addObjs` [Array](#)
- `index` [Number](#)

indexOf

Exact same function as `Array.prototype.indexOf`.

HACK: ugly hack for Baidu mobile browser compatibility, stupid Baidu guys modify `Array.prototype.indexOf` for all pages loaded, their version changes strict comparison to non-strict comparison, it also ignores the second parameter of the original API, and this will cause event handler enter infinite loop.

Baidu developers, if you ever see this documentation, here is the

standard: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/indexOf, Seriously!

| meta | description |
|------------|---|
| Returns | Number |
| Defined in | cocos2d/core/platform/js.js:786 |

Parameters

- `searchElement` Any Element to locate in the array.
- `fromIndex` [Number](#) The index to start the search at

contains

Determines whether the array contains a specific value.

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/js.js:798 |

Parameters

- `array Any`
- `value Any`

copy

Copy an array's item to a new array (its performance is better than Array.slice)

| meta | description |
|------------|---|
| Returns | Array |
| Defined in | cocos2d/core/platform/js.js:809 |

Parameters

- `array Array`

Asset Class

Extends [RawAsset](#)

Defined in: <https://github.com/cocos-creator/engine/blob/4f734a806d1fd7c4073fb064fddc961384fe67af/cocos2d/core/assets/CCAsset.js:32>

Module: [cc](#)

Base class for handling assets used in Creator.

You may want to override:

- `createNode`
- `getset functions of _nativeAsset`
- `cc.Object._serialize`

- cc.Object._deserialize

Index

Properties

- `loaded` Boolean Whether the asset is loaded or not
- `nativeUrl` String Returns the url of this asset's native object, if none it will returns an empty string.
- `_native` String Serializable url for native asset.
- `_nativeAsset` Object The underlying native asset of this asset if one is available.
- `preventDeferredLoadDependents` Boolean Indicates whether its dependent raw assets can support deferred load if the owner scene (or prefab) is marked as `asyncLoadAssets`.
- `preventPreloadNativeObject` Boolean Indicates whether its native object should be preloaded from native url.
- `_uuid` String
- `_name` String
- `_objFlags` Number
- `name` String The name of the object.
- `isValid` Boolean Indicates whether the object is not yet destroyed.

Methods

- `deserialize` 应 AssetDB 要求提供这个方法
- `toString` Returns the asset's url.
- `serialize` 应 AssetDB 要求提供这个方法
- `createNode` Create a new node using this asset in the scene....
- `_setRawAsset` Set native file name for this asset.
- `destroy` Actual object destruction will delayed until before rendering.
- `_destruct` Clear all references in the instance.
- `_onPreDestroy` Called before the object being destroyed.
- `_serialize` The customized serialization for this object.
- `_deserialize` Init this object from the custom serialized data.

Details

Properties

loaded

Whether the asset is loaded or not

| meta | description |
|------|-------------------------|
| Type | Boolean |

| meta | description |
|-------------|---|
| Defined in | cocos2d/core/assets/CCAsset.js:57 |

nativeUrl

Returns the url of this asset's native object, if none it will returns an empty string.

| meta | description |
|-------------|---|
| Type | String |
| Defined in | cocos2d/core/assets/CCAsset.js:70 |

_native

Serializable url for native asset.

| meta | description |
|-------------|--|
| Type | String |
| Defined in | cocos2d/core/assets/CCAsset.js:108 |

_nativeAsset

The underlying native asset of this asset if one is available. This property can be used to access additional details or functionality related to the asset. This property will be initialized by the loader if _native is available.

| meta | description |
|-------------|--|
| Type | Object |
| Defined in | cocos2d/core/assets/CCAsset.js:116 |

preventDeferredLoadDependents

Indicates whether its dependent raw assets can support deferred load if the owner scene (or prefab) is marked as `asyncLoadAssets`.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/assets/CCAsset.js:144 |

preventPreloadNativeObject

Indicates whether its native object should be preloaded from native url.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/assets/CCAsset.js:154 |

_uuid

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/assets/CCRawAsset.js:46 |

_name

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:76 |

_objFlags

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/platform/CCObject.js:83 |

name

The name of the object.

| meta | description |
|------------|---|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:243 |

Examples

```
obj.name = "New Obj";
```

isValid

Indicates whether the object is not yet destroyed. (It will not be available after being destroyed)
When an object's `destroy` is called, it is actually destroyed after the end of this frame.
So `isValid` will return false from the next frame, while `isValid` in the current frame will still be true.
If you want to determine whether the current frame has called `destroy`, use `cc.isValid(obj, true)`, but this is often caused by a particular logical requirements, which is not normally required.

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:261 |

Examples

```
var node = new cc.Node();
cc.log(node.isValid);    // true
node.destroy();
cc.log(node.isValid);    // true, still valid in this frame
// after a frame...
cc.log(node.isValid);    // false, destroyed in the end of last frame
```

Methods

deserialize

应 AssetDB 要求提供这个方法

| meta | description |
|------------|--|
| Returns | Asset |
| Defined in | cocos2d/core/assets/CCAsset.js:131 |

Parameters

- data [String](#)

toString

Returns the asset's url.

The Asset object overrides the `toString()` method of the `Object` object. For Asset objects, the `toString()` method returns a string representation of the object. JavaScript calls the `toString()` method automatically when an asset is to be represented as a text value or when a texture is referred to in a string concatenation.

| meta | description |
|------------|--|
| Returns | String |
| Defined in | cocos2d/core/assets/CCAsset.js:165 |

serialize

应 AssetDB 要求提供这个方法

| meta | description |
|------------|--|
| Returns | String |
| Defined in | cocos2d/core/assets/CCAsset.js:179 |

createNode

Create a new node using this asset in the scene.

If this type of asset dont have its corresponding node type, this method should be null.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/assets/CCAsset.js:190 |

Parameters

- `callback` [Function](#)
- `error` [String](#) null or the error info
- `node` [Object](#) the created node or null

_setRawAsset

Set native file name for this asset.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/assets/CCAsset.js:205 |

Parameters

- `filename` [String](#)
- `inLibrary` [Boolean](#)

destroy

Destroy this Object, and release all its own references to other objects.

Actual object destruction will delayed until before rendering. From the next frame , this object is not usable any more. You can use cc.isValid(obj) to check whether the object is destroyed before accessing it.

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:296 |

Examples

```
obj.destroy();
```

_destruct

Clear all references in the instance.

NOTE: this method will not clear the getter or setter functions which defined in the instance of CCOObject. You can override the _destruct method if you need, for example:

```
_destruct: function () { for (var key in this) { if (this.hasOwnProperty(key)) { switch (typeof this[key]) { case 'string': this[key] = ''; break; case 'object': case 'function': this[key] = null; break; } } }}
```

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:430 |

_onPreDestroy

Called before the object being destroyed.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:463 |

_serialize

The customized serialization for this object. (Editor Only)

| meta | description |
|------------|---|
| Returns | object |
| Defined in | cocos2d/core/platform/CCObject.js:488 |

Parameters

- exporting [Boolean](#)

_deserialize

Init this object from the custom serialized data.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:498 |

Parameters

- `data Object` the serialized json data
- `ctx _Deserializer`

AudioClip Class

Extends [Asset](#), [EventTarget](#)(mixin)

Module: [cc](#)

Class for audio data handling.

Index

Properties

- `loaded Boolean` Whether the asset is loaded or not
- `nativeUrl String` Returns the url of this asset's native object, if none it will returns an empty string.
- `_native String` Serializable url for native asset.
- `_nativeAsset Object` The underlying native asset of this asset if one is available.
- `_uuid String`
- `_name String`
- `_objFlags Number`
- `name String` The name of the object.
- `isValid Boolean` Indicates whether the object is not yet destroyed.

Methods

- `toString` Returns the asset's url.
- `serialize` 应 AssetDB 要求提供这个方法
- `createNode` Create a new node using this asset in the scene....
- `_setRawAsset` Set native file name for this asset.
- `hasEventListener` Checks whether the EventTarget object has any callback registered for a specific type of event.
- `on` Register an callback of a specific event type on the EventTarget.
- `off` Removes the listeners previously registered with the same type, callback, target and or useCapture,...
- `targetOff` Removes all callbacks previously registered with the same target (passed as parameter).
- `once` Register an callback of a specific event type on the EventTarget,...

- `emit` Trigger an event directly with the event name and necessary arguments.
- `dispatchEvent` Send an event with the event object.
- `destroy` Actual object destruction will delayed until before rendering.
- `_destruct` Clear all references in the instance.
- `_onPreDestroy` Called before the object being destroyed.
- `_serialize` The customized serialization for this object.
- `_deserialize` Init this object from the custom serialized data.

Events

- `load` This event is emitted when the asset is loaded

Details

Properties

loaded

Whether the asset is loaded or not

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/assets/CCAsset.js:57 |

nativeUrl

Returns the url of this asset's native object, if none it will returns an empty string.

| meta | description |
|------------|---|
| Type | String |
| Defined in | cocos2d/core/assets/CCAsset.js:70 |

_native

Serializable url for native asset.

| meta | description |
|-------------|--|
| Type | String |
| Defined in | cocos2d/core/assets/CCAsset.js:108 |

_nativeAsset

The underlying native asset of this asset if one is available. This property can be used to access additional details or functionality related to the asset. This property will be initialized by the loader if `_native` is available.

| meta | description |
|-------------|--|
| Type | Object |
| Defined in | cocos2d/core/assets/CCAsset.js:116 |

_uuid

| meta | description |
|-------------|--|
| Type | String |
| Defined in | cocos2d/core/assets/CCRawAsset.js:46 |

_name

| meta | description |
|-------------|--|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:76 |

_objFlags

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/platform/CCObject.js:83 |

name

The name of the object.

| meta | description |
|------------|---|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:243 |

Examples

```
obj.name = "New Obj";
```

isValid

Indicates whether the object is not yet destroyed. (It will not be available after being destroyed)

When an object's `destroy` is called, it is actually destroyed after the end of this frame. So `isValid` will return false from the next frame, while `isValid` in the current frame will still be true. If you want to determine whether the current frame has called `destroy`, use `cc.isValid(obj, true)`, but this is often caused by a particular logical requirements, which is not normally required.

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:261 |

Examples

```
var node = new cc.Node();
cc.log(node.isValid); // true
```

```
node.destroy();
cc.log(node.isValid);      // true, still valid in this frame
// after a frame...
cc.log(node.isValid);      // false, destroyed in the end of last frame
```

Methods

toString

Returns the asset's url.

The `Asset` object overrides the `toString()` method of the `Object` object.
For `Asset` objects, the `toString()` method returns a string representation of the object.
JavaScript calls the `toString()` method automatically when an asset is to be
represented as a text value or when a texture is referred to in a string concatenation.

| meta | description |
|------------|--|
| Returns | String |
| Defined in | cocos2d/core/assets/CCAsset.js:165 |

serialize

应 AssetDB 要求提供这个方法

| meta | description |
|------------|--|
| Returns | String |
| Defined in | cocos2d/core/assets/CCAsset.js:179 |

createNode

Create a new node using this asset in the scene.
If this type of asset dont have its corresponding node type, this method should be
null.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/assets/CCAsset.js:190 |

Parameters

- callback [Function](#)
- error [String](#) null or the error info
- node [Object](#) the created node or null

_setRawAsset

Set native file name for this asset.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/assets/CCAsset.js:205 |

Parameters

- filename [String](#)
- inLibrary [Boolean](#)

hasEventListener

Checks whether the EventTarget object has any callback registered for a specific type of event.

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/event/event-target.js:68 |

Parameters

- type [String](#) The type of event.

on

Register an callback of a specific event type on the EventTarget. This type of event should be triggered via `emit`.

| meta | description |
|---------|--------------------------|
| Returns | Function |

| meta | description |
|------------|---|
| Defined in | cocos2d/core/event/event-target.js:76 |

Parameters

- type [String](#) A string representing the event type to listen for.
- callback [Function](#) The callback that will be invoked when the event is dispatched.
- The callback is ignored if it is a duplicate (the callbacks are unique).
- arg1 Any arg1
- arg2 Any arg2
- arg3 Any arg3
- arg4 Any arg4
- arg5 Any arg5
- target [Object](#) The target (this object) to invoke the callback, can be null

Examples

```
eventTarget.on('fire', function (event) {
    cc.log("fire in the hole");
}, node);
```

off

Removes the listeners previously registered with the same type, callback, target and or useCapture, if only type is passed as parameter, all listeners registered with that type will be removed.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/event/event-target.js:117 |

Parameters

- type [String](#) A string representing the event type being removed.
- callback [Function](#) The callback to remove.
- target [Object](#) The target (this object) to invoke the callback, if it's not given, only callback without target will be removed

Examples

```
// register fire eventListener
var callback = eventTarget.on('fire', function (event) {
    cc.log("fire in the hole");
}, target);
```

```
// remove fire event listener
eventTarget.off('fire', callback, target);
// remove all fire event listeners
eventTarget.off('fire');
```

targetOff

Removes all callbacks previously registered with the same target (passed as parameter). This is not for removing all listeners in the current event target, and this is not for removing all listeners the target parameter have registered. It's only for removing all listeners (callback and target couple) registered on the current event target by the target parameter.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/event/event-target.js:151 |

Parameters

- **target** [Object](#) The target to be searched for all related listeners

once

Register an callback of a specific event type on the EventTarget, the callback will remove itself after the first time it is triggered.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/event/event-target.js:164 |

Parameters

- **type** [String](#) A string representing the event type to listen for.
- **callback** [Function](#) The callback that will be invoked when the event is dispatched.
 - The callback is ignored if it is a duplicate (the callbacks are unique).
 - **arg1** Any arg1
 - **arg2** Any arg2
 - **arg3** Any arg3
 - **arg4** Any arg4
 - **arg5** Any arg5
- **target** [Object](#) The target (this object) to invoke the callback, can be null

Examples

```
eventTarget.once('fire', function (event) {
    cc.log("this is the callback and will be invoked only once");
}, node);
```

emit

Trigger an event directly with the event name and necessary arguments.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/event/event-target.js:201 |

Parameters

- type [String](#) event type
- arg1 Any First argument
- arg2 Any Second argument
- arg3 Any Third argument
- arg4 Any Fourth argument
- arg5 Any Fifth argument

Examples

```
eventTarget.emit('fire', event);
eventTarget.emit('fire', message, emitter);
```

dispatchEvent

Send an event with the event object.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/event/event-target.js:221 |

Parameters

- event [Event](#)

destroy

Destroy this Object, and release all its own references to other objects.
Actual object destruction will delayed until before rendering. From the next frame, this

object is not usable any more. You can use cc.isValid(obj) to check whether the object is destroyed before accessing it.

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:296 |

Examples

```
obj.destroy();
```

_destruct

Clear all references in the instance.

NOTE: this method will not clear the getter or setter functions which defined in the instance of CCObject. You can override the _destruct method if you need, for example:

```
_destruct: function () { for (var key in this) { if (this.hasOwnProperty(key)) { switch (typeof this[key]) { case 'string': this[key] = ""; break; case 'object': case 'function': this[key] = null; break; } } }
```

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:430 |

_onPreDestroy

Called before the object being destroyed.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:463 |

_serialize

The customized serialization for this object. (Editor Only)

| meta | description |
|------------|---|
| Returns | object |
| Defined in | cocos2d/core/platform/CCObject.js:488 |

Parameters

- `exporting Boolean`

`_deserialize`

Init this object from the custom serialized data.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:498 |

Parameters

- `data Object` the serialized json data
- `ctx _Deserializer`

Events

load Event

Module: [cc](#)

This event is emitted when the asset is loaded

audioEngine Class

Module: [cc](#)

`cc.audioEngine` is the singleton object, it provide simple audio APIs.

Index

Methods

- `play` Play audio.
- `setLoop` Set audio loop.
- `isLoop` Get audio cycle state.

- `setVolume` Set the volume of audio.
- `getVolume` The volume of the music max value is 1.0, the min value is 0.0 .
- `setCurrentTime` Set current time
- `getCurrentTime` Get current time
- `getDuration` Get audio duration
- `getState` Get audio state
- `setFinishCallback` Set Audio finish callback
- `pause` Pause playing audio.
- `pauseAll` Pause all playing audio
- `resume` Resume playing audio.
- `resumeAll` Resume all playing audio.
- `stop` Stop playing audio.
- `stopAll` Stop all playing audio.
- `setMaxAudioInstance` Set up an audio can generate a few examples.
- `getMaxAudioInstance` Getting audio can produce several examples.
- `uncache` Unload the preloaded audio from internal buffer.
- `uncacheAll` Unload all audio from internal buffer.
- `preload` Preload audio file.
- `setMaxWebAudioSize` Set a size, the unit is KB.
- `playMusic` Play background music
- `stopMusic` Stop background music.
- `pauseMusic` Pause the background music.
- `resumeMusic` Resume playing background music.
- `getMusicVolume` Get the volume(0.0 ~ 1.0).
- `setMusicVolume` Set the background music volume.
- `isMusicPlaying` Background music playing state
- `playEffect` Play effect audio.
- `setEffectsVolume` Set the volume of effect audio.
- `getEffectsVolume` The volume of the effect audio max value is 1.0, the min value is 0.0 .
- `pauseEffect` Pause effect audio.
- `pauseAllEffects` Stop playing all the sound effects.
- `resumeEffect` Resume effect audio.
- `resumeAllEffects` Resume all effect audio.
- `stopEffect` Stop playing the effect audio.
- `stopAllEffects` Stop playing all the effects.

Details

Methods

`play`

Play audio.

| meta | description |
|---------|------------------------|
| Returns | Number |

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:114 |

Parameters

- `clip` [AudioClip](#) The audio clip to play.
- `loop` [Boolean](#) Whether the music loop or not.
- `volume` [Number](#) Volume size.

Examples

```
cc.loader.loadRes(url, cc.AudioClip, function (err, clip) {
    var audioID = cc.audioEngine.play(clip, false, 0.5);
});
```

setLoop

Set audio loop.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:161 |

Parameters

- `audioID` [Number](#) audio id.
- `loop` [Boolean](#) Whether cycle.

Examples

```
cc.audioEngine.setLoop(id, true);
```

isLoop

Get audio cycle state.

| meta | description |
|------------|--|
| Returns | Boolean |
| Defined in | cocos2d/audio/CCAudioEngine.js:177 |

Parameters

- `audioID` [Number](#) audio id.

Examples

```
cc.audioEngine.isLoop(id);
```

setVolume

Set the volume of audio.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:193 |

Parameters

- `audioID` [Number](#) audio id.
- `volume` [Number](#) Volume must be in 0.0~1.0 .

Examples

```
cc.audioEngine.setVolume(id, 0.5);
```

getVolume

The volume of the music max value is 1.0,the min value is 0.0 .

| meta | description |
|------------|--|
| Returns | Number |
| Defined in | cocos2d/audio/CCAudioEngine.js:209 |

Parameters

- `audioID` [Number](#) audio id.

Examples

```
var volume = cc.audioEngine.getVolume(id);
```

setcurrentTime

Set current time

| meta | description |
|------------|--|
| Returns | Boolean |
| Defined in | cocos2d/audio/CCAudioEngine.js:223 |

Parameters

- `audioID` [Number](#) audio id.
- `sec` [Number](#) current time.

Examples

```
cc.audioEngine.setCurrentTime(id, 2);
```

getCurrentTime

Get current time

| meta | description |
|------------|--|
| Returns | Number |
| Defined in | cocos2d/audio/CCAudioEngine.js:244 |

Parameters

- `audioID` [Number](#) audio id.

Examples

```
var time = cc.audioEngine.getCurrentTime(id);
```

getDuration

Get audio duration

| meta | description |
|------------|--|
| Returns | Number |
| Defined in | cocos2d/audio/CCAudioEngine.js:258 |

Parameters

- `audioID` [Number](#) audio id.

Examples

```
var time = cc.audioEngine.getDuration(id);
```

getState

Get audio state

| meta | description |
|------------|--|
| Returns | audioEngine.AudioState |
| Defined in | cocos2d/audio/CCAudioEngine.js:272 |

Parameters

- `audioID` [Number](#) audio id.

Examples

```
var state = cc.audioEngine.getState(id);
```

setFinishCallback

Set Audio finish callback

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:286 |

Parameters

- `audioID` [Number](#) audio id.
- `callback` [Function](#) loaded callback.

Examples

```
cc.audioEngine.setFinishCallback(id, function () {});
```

`pause`

Pause playing audio.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:302 |

Parameters

- `audioID` [Number](#) The return value of function play.

Examples

```
cc.audioEngine.pause(audioID);
```

`pauseAll`

Pause all playing audio

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:322 |

Examples

```
cc.audioEngine.pauseAll();
```

`resume`

Resume playing audio.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:340 |

Parameters

- `audioID` [Number](#) The return value of function play.

Examples

```
cc.audioEngine.resume(audioID);
```

resumeAll

Resume all playing audio.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:355 |

Examples

```
cc.audioEngine.resumeAll();
```

stop

Stop playing audio.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:372 |

Parameters

- `audioID` [Number](#) The return value of function play.

Examples

```
cc.audioEngine.stop(audioID);
```

stopAll

Stop all playing audio.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:392 |

Examples

```
cc.audioEngine.stopAll();
```

setMaxAudioInstance

Set up an audio can generate a few examples.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:409 |

Parameters

- num [Number](#) a number of instances to be created from within an audio

Examples

```
cc.audioEngine.setMaxAudioInstance(20);
```

getMaxAudioInstance

Getting audio can produce several examples.

| meta | description |
|------------|--|
| Returns | Number |
| Defined in | cocos2d/audio/CCAudioEngine.js:421 |

Examples

```
cc.audioEngine.getMaxAudioInstance();
```

uncache

Unload the preloaded audio from internal buffer.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:433 |

Parameters

- `clip` [AudioClip](#)

Examples

```
cc.audioEngine.uncache(filePath);
```

uncacheAll

Unload all audio from internal buffer.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:468 |

Examples

```
cc.audioEngine.uncacheAll();
```

preload

Preload audio file.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:499 |
| Deprecated | `cc.audioEngine.preload` is deprecated, use `cc.loader.loadRes(url, cc.AudioClip)` instead please. |

Parameters

- `filePath` [String](#) The file path of an audio.
- `callback` [Function](#) The callback of an audio.

Examples

```
cc.audioEngine.preload(path);
```

setMaxWebAudioSize

Set a size, the unit is KB. Over this size is directly resolved into DOM nodes.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:521 |

Parameters

- `kb` [Number](#) The file path of an audio.

Examples

```
cc.audioEngine.setMaxWebAudioSize(300);
```

playMusic

Play background music

| meta | description |
|------------|--|
| Returns | Number |
| Defined in | cocos2d/audio/CCAudioEngine.js:573 |

Parameters

- `clip` [AudioClip](#) The audio clip to play.
- `loop` [Boolean](#) Whether the music loop or not.

Examples

```
cc.loader.loadRes(url, cc.AudioClip, function (err, clip) {
    var audioID = cc.audioEngine.playMusic(clip, false);
});
```

stopMusic

Stop background music.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:593 |

Examples

```
cc.audioEngine.stopMusic();
```

pauseMusic

Pause the background music.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:604 |

Examples

```
cc.audioEngine.pauseMusic();
```

resumeMusic

Resume playing background music.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:616 |

Examples

```
cc.audioEngine.resumeMusic();
```

getMusicVolume

Get the volume(0.0 ~ 1.0).

| meta | description |
|------------|--|
| Returns | Number |
| Defined in | cocos2d/audio/CCAudioEngine.js:628 |

Examples

```
var volume = cc.audioEngine.getMusicVolume();
```

setMusicVolume

Set the background music volume.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:640 |

Parameters

- `volume` [Number](#) Volume must be in 0.0~1.0.

Examples

```
cc.audioEngine.setMusicVolume(0.5);
```

isMusicPlaying

Background music playing state

| meta | description |
|------------|--|
| Returns | Boolean |
| Defined in | cocos2d/audio/CCAudioEngine.js:655 |

Examples

```
cc.audioEngine.isMusicPlaying();
```

playEffect

Play effect audio.

| meta | description |
|------------|--|
| Returns | Number |
| Defined in | cocos2d/audio/CCAudioEngine.js:667 |

Parameters

- `clip` [AudioClip](#) The audio clip to play.
- `loop` [Boolean](#) Whether the music loop or not.

Examples

```
cc.loader.loadRes(url, cc.AudioClip, function (err, clip) {  
    var audioID = cc.audioEngine.playEffect(clip, false);  
});
```

setEffectsVolume

Set the volume of effect audio.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:683 |

Parameters

- `volume` [Number](#) Volume must be in 0.0~1.0.

Examples

```
cc.audioEngine.setEffectsVolume(0.5);
```

getEffectsVolume

The volume of the effect audio max value is 1.0, the min value is 0.0 .

| meta | description |
|------------|--|
| Returns | Number |
| Defined in | cocos2d/audio/CCAudioEngine.js:700 |

Examples

```
var volume = cc.audioEngine.getEffectsVolume();
```

pauseEffect

Pause effect audio.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:712 |

Parameters

- `audioID` [Number](#) audio id.

Examples

```
cc.audioEngine.pauseEffect(audioID);
```

pauseAllEffects

Stop playing all the sound effects.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:724 |

Examples

```
cc.audioEngine.pauseAllEffects();
```

resumeEffect

Resume effect audio.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:747 |

Parameters

- `audioID` [Number](#) The return value of function play.

Examples

```
cc.audioEngine.resumeEffect(audioID);
```

resumeAllEffects

Resume all effect audio.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:759 |

Examples

```
cc.audioEngine.resumeAllEffects();
```

stopEffect

Stop playing the effect audio.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:776 |

Parameters

- `audioID` [Number](#) audio id.

Examples

```
cc.audioEngine.stopEffect(id);
```

stopAllEffects

Stop playing all the effects.

| meta | description |
|------------|--|
| Defined in | cocos2d/audio/CCAudioEngine.js:788 |

Examples

```
cc.audioEngine.stopAllEffects();
```

AudioSource Class

Extends [Component](#)

Module: [cc](#)

Audio Source.

Index

Properties

- `isPlaying` Boolean Is the audio source playing (Read Only).
- `clip` AudioClip The clip of the audio source to play.
- `volume` Number The volume of the audio source.
- `mute` Boolean Is the audio source mute?
- `loop` Boolean Is the audio source looping?
- `playOnLoad` Boolean If set to true, the audio source will automatically start playing on onEnable.
- `__eventTargets` Array Register all related EventTargets,...
- `node` Node The node this component is attached to.
- `uuid` String The uuid for editor.
- `_enabled` Boolean
- `enabled` Boolean indicates whether this component is enabled or not.
- `enabledInHierarchy` Boolean indicates whether this component is enabled and its node is also active in the hierarchy.
- `_isOnLoadCalled` Number Returns a value which used to indicate the onLoad get called or not.
- `_name` String
- `_objFlags` Number
- `name` String The name of the object.
- `isValid` Boolean Indicates whether the object is not yet destroyed.

Methods

- `play` Plays the clip.
- `stop` Stops the clip.
- `pause` Pause the clip.
- `resume` Resume the clip.
- `rewind` Rewind playing music.
- `getCurrentTime` Get current time
- `setCurrentTime` Set current time
- `getDuration` Get audio duration
- `update` This is a lifecycle method.
- `lateUpdate` This is a lifecycle method.
- `__preload` __preload is called before every onLoad.
- `onLoad` When attaching to an active node or its node first activated.
- `start` Called before all scripts' update if the Component is enabled the first time.
- `onEnable` This is a lifecycle method.
- `onDisable` This is a lifecycle method.
- `onDestroy` This is a lifecycle method.
- `onFocusInEditor`
- `onLostFocusInEditor`
- `resetInEditor` Called to initialize the component or node's properties when adding the component the first time or when the Reset command is used.
- `addComponent` Adds a component class to the node.

- `getComponent` Returns the component of supplied type if the node has one attached, null if it doesn't....
- `getComponents` Returns all components of supplied Type in the node.
- `getComponentInChildren` Returns the component of supplied type in any of its children using depth first search.
- `getComponentsInChildren` Returns the components of supplied type in self or any of its children using depth first search.
- `_getLocalBounds` If the component's bounding box is different from the node's, you can implement this method to supply
- `onRestore` for undo/redo operation.
- `schedule` Schedules a custom selector....
- `scheduleOnce` Schedules a callback function that runs only once, with a delay of 0 or larger.
- `unschedule` Unschedules a custom callback function.
- `unscheduleAllCallbacks` unschedule all scheduled callback functions: custom callback functions, and the 'update' callback function....
- `destroy` Actual object destruction will delayed until before rendering.
- `_destruct` Clear all references in the instance.
- `_onPreDestroy` Called before the object being destroyed.
- `_serialize` The customized serialization for this object.
- `_deserialize` Init this object from the custom serialized data.

Details

Properties

isPlaying

Is the audio source playing (Read Only).

Note: isPlaying is not supported for Native platforms.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CC AudioSource.js:68 |

clip

The clip of the audio source to play.

| meta | description |
|------|---------------------------|
| Type | AudioClip |

| meta | description |
|-------------|---|
| Defined in | cocos2d/core/components/CCAudioSource.js:88 |

volume

The volume of the audio source.

| meta | description |
|-------------|--|
| Type | Number |
| Defined in | cocos2d/core/components/CCAudioSource.js:126 |

mute

Is the audio source mute?

| meta | description |
|-------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCAudioSource.js:148 |

loop

Is the audio source looping?

| meta | description |
|-------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCAudioSource.js:168 |

playOnLoad

If set to true, the audio source will automatically start playing on onEnable.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCAudioSource.js:188 |

__eventTargets

Register all related EventTargets, all event callbacks will be removed in _onPreDestroy

| meta | description |
|------------|---|
| Type | Array |
| Defined in | cocos2d/core/components/CCComponent.js:61 |

node

The node this component is attached to. A component is always attached to a node.

| meta | description |
|------------|---|
| Type | Node |
| Defined in | cocos2d/core/components/CCComponent.js:75 |

Examples

```
cc.log(comp.node);
```

uuid

The uuid for editor.

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/components/CCComponent.js:106 |

Examples

```
cc.log(comp.uuid);
```

_enabled

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:147 |

enabled

indicates whether this component is enabled or not.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:154 |

Examples

```
comp.enabled = true;  
cc.log(comp.enabled);
```

enabledInHierarchy

indicates whether this component is enabled and its node is also active in the hierarchy.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:185 |

Examples

```
cc.log(comp.enabledInHierarchy);
```

_isOnLoadCalled

Returns a value which used to indicate the onLoad get called or not.

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/components/CCComponent.js:201 |

Examples

```
cc.log(this._isOnLoadCalled > 0);
```

_name

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:76 |

_objFlags

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/platform/CCObject.js:83 |

name

The name of the object.

| meta | description |
|------|------------------------|
| Type | String |

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:243 |

Examples

```
obj.name = "New Obj";
```

isValid

Indicates whether the object is not yet destroyed. (It will not be available after being destroyed)
When an object's `destroy` is called, it is actually destroyed after the end of this frame.
So `isValid` will return false from the next frame, while `isValid` in the current frame will still be true.
If you want to determine whether the current frame has called `destroy`, use `cc.isValid(obj, true)`, but this is often caused by a particular logical requirements, which is not normally required.

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:261 |

Examples

```
var node = new cc.Node();
cc.log(node.isValid);      // true
node.destroy();
cc.log(node.isValid);      // true, still valid in this frame
// after a frame...
cc.log(node.isValid);      // false, destroyed in the end of last frame
```

Methods

play

Plays the clip.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCAudioSource.js:255 |

stop

Stops the clip.

| meta | description |
|-------------|--|
| Defined in | cocos2d/core/components/CCAudioSource.js:272 |

pause

Pause the clip.

| meta | description |
|-------------|--|
| Defined in | cocos2d/core/components/CCAudioSource.js:281 |

resume

Resume the clip.

| meta | description |
|-------------|--|
| Defined in | cocos2d/core/components/CCAudioSource.js:290 |

rewind

Rewind playing music.

| meta | description |
|-------------|--|
| Defined in | cocos2d/core/components/CCAudioSource.js:300 |

getCurrentTime

Get current time

| meta | description |
|-------------|--|
| Returns | Number |
| Defined in | cocos2d/core/components/CCAudioSource.js:309 |

setCurrentTime

Set current time

| meta | description |
|------------|--|
| Returns | Number |
| Defined in | cocos2d/core/components/CCAudioSource.js:319 |

Parameters

- `time` [Number](#)

getDuration

Get audio duration

| meta | description |
|------------|--|
| Returns | Number |
| Defined in | cocos2d/core/components/CCAudioSource.js:331 |

update

Update is called every frame, if the Component is enabled.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:222 |

Parameters

- `dt` [Number](#) the delta time in seconds it took to complete the last frame

lateUpdate

LateUpdate is called every frame, if the Component is enabled.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:233 |

__preload

__preload is called before every onLoad. It is used to initialize the builtin components internally, to avoid checking whether onLoad is called before every public method calls. This method should be removed if script priority is supported.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:243 |

onLoad

When attaching to an active node or its node first activated. onLoad is always called before any start functions, this allows you to order initialization of scripts.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:254 |

start

Called before all scripts' update if the Component is enabled the first time. Usually used to initialize some logic which need to be called after all components' onload methods called.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:267 |

onEnable

Called when this component becomes enabled and its node is active.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:280 |

onDisable

Called when this component becomes disabled or its node becomes inactive.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:290 |

onDestroy

Called when this component will be destroyed.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:300 |

onFocusInEditor

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:310 |

onLostFocusInEditor

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:315 |

resetInEditor

Called to initialize the component or node's properties when adding the component the first time or when the Reset command is used. This function is only called in editor.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:320 |

addComponent

Adds a component class to the node. You can also add component to node by passing in the name of the script.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:330 |

Parameters

- typeOrClassName [Function](#) | [String](#) the constructor or the class name of the component to add

Examples

```
var sprite = node.addComponent(cc.Sprite);
var test = node.addComponent("Test");
```

getComponent

Returns the component of supplied type if the node has one attached, null if it doesn't. You can also get component in the node by passing in the name of the script.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:348 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
// get sprite component.
var sprite = node.getComponent(cc.Sprite);
// get custom test calss.
var test = node.getComponent("Test");
```

getComponents

Returns all components of supplied Type in the node.

| meta | description |
|------------|--|
| Returns | Component[] |
| Defined in | cocos2d/core/components/CCComponent.js:372 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
var sprites = node.getComponents(cc.Sprite);
var tests = node.getComponents("Test");
```

getComponentInChildren

Returns the component of supplied type in any of its children using depth first search.

| meta | description |
|---------|---------------------------|
| Returns | Component |

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:390 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
var sprite = node.getComponentInChildren(cc.Sprite);
var Test = node.getComponentInChildren("Test");
```

getComponentsInChildren

Returns the components of supplied type in self or any of its children using depth first search.

| meta | description |
|------------|--|
| Returns | Component[] |
| Defined in | cocos2d/core/components/CCComponent.js:408 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
var sprites = node.getComponentsInChildren(cc.Sprite);
var tests = node.getComponentsInChildren("Test");
```

_getLocalBounds

If the component's bounding box is different from the node's, you can implement this method to supply a custom axis aligned bounding box (AABB), so the editor's scene view can perform hit test properly.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:428 |

Parameters

- `out_rect Rect` the Rect to receive the bounding box

onRestore

onRestore is called after the user clicks the Reset item in the Inspector's context menu or performs an undo operation on this component.

If the component contains the "internal state", short for "temporary member variables which not included in its CCClass properties", then you may need to implement this function.

The editor will call the getset accessors of your component to record/restore the component's state for undo/redo operation. However, in extreme cases, it may not work well. Then you should implement this function to manually synchronize your component's "internal states" with its public properties. Once you implement this function, all the getset accessors of your component will not be called when the user performs an undo/redo operation. Which means that only the properties with default value will be recorded or restored by editor.

Similarly, the editor may fail to reset your component correctly in extreme cases. Then if you need to support the reset menu, you should manually synchronize your component's "internal states" with its properties in this function. Once you implement this function, all the getset accessors of your component will not be called during reset operation. Which means that only the properties with default value will be reset by editor.

This function is only called in editor mode.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:441 |

schedule

Schedules a custom selector.

If the selector is already scheduled, then the interval parameter will be updated without scheduling it again.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:535 |

Parameters

- `callback` [function](#) The callback function
- `interval` [Number](#) Tick interval in seconds. 0 means tick every frame.
- `repeat` [Number](#) The selector will be executed (repeat + 1) times, you can use cc.macro.REPEAT_FOREVER for tick infinitely.
- `delay` [Number](#) The amount of time that the first tick will wait before execution.

Examples

```
var timeCallback = function (dt) {
    cc.log("time: " + dt);
}
this.schedule(timeCallback, 1);
```

scheduleOnce

Schedules a callback function that runs only once, with a delay of 0 or larger.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:572 |

Parameters

- `callback` [function](#) A function wrapped as a selector
- `delay` [Number](#) The amount of time that the first tick will wait before execution.

Examples

```
var timeCallback = function (dt) {
    cc.log("time: " + dt);
}
this.scheduleOnce(timeCallback, 2);
```

unschedule

Unschedules a custom callback function.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:589 |

Parameters

- `callback_fn` [function](#) A function wrapped as a selector

Examples

```
this.unschedule(_callback);
```

unscheduleAllCallbacks

unschedule all scheduled callback functions: custom callback functions, and the 'update' callback function.

Actions are not affected by this method.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:605 |

Examples

```
this.unscheduleAllCallbacks();
```

destroy

Destroy this Object, and release all its own references to other objects.

Actual object destruction will delayed until before rendering. From the next frame, this object is not usable any more. You can use `cc.isValid(obj)` to check whether the object is destroyed before accessing it.

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:296 |

Examples

```
obj.destroy();
```

_destruct

Clear all references in the instance.

NOTE: this method will not clear the getter or setter functions which defined in the instance of CCObject. You can override the _destruct method if you need, for example: _destruct: function () { for (var key in this) { if (this.hasOwnProperty(key)) { switch (typeof this[key]) { case 'string': this[key] = ""; break; case 'object': case 'function': this[key] = null; break; } } }

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:430 |

_onPreDestroy

Called before the object being destroyed.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:463 |

_serialize

The customized serialization for this object. (Editor Only)

| meta | description |
|------------|---|
| Returns | object |
| Defined in | cocos2d/core/platform/CCObject.js:488 |

Parameters

- exporting [Boolean](#)

_deserialize

Init this object from the custom serialized data.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:498 |

Parameters

- `data Object` the serialized json data
- `ctx _Deserializer`

BitmapFont Class

Extends [Font](#)

Module: [cc](#)

Class for BitmapFont handling.

Index

Properties

- `loaded Boolean` Whether the asset is loaded or not
- `nativeUrl String` Returns the url of this asset's native object, if none it will returns an empty string.
- `_native String` Serializable url for native asset.
- `_nativeAsset Object` The underlying native asset of this asset if one is available.
- `_uuid String`
- `_name String`
- `_objFlags Number`
- `name String` The name of the object.
- `isValid Boolean` Indicates whether the object is not yet destroyed.

Methods

- `toString` Returns the asset's url.
- `serialize` 应 AssetDB 要求提供这个方法
- `createNode` Create a new node using this asset in the scene....
- `_setRawAsset` Set native file name for this asset.
- `destroy` Actual object destruction will delayed until before rendering.
- `_destruct` Clear all references in the instance.
- `_onPreDestroy` Called before the object being destroyed.
- `_serialize` The customized serialization for this object.
- `_deserialize` Init this object from the custom serialized data.

Details

Properties

loaded

Whether the asset is loaded or not

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/assets/CCAsset.js:57 |

nativeUrl

Returns the url of this asset's native object, if none it will returns an empty string.

| meta | description |
|------------|---|
| Type | String |
| Defined in | cocos2d/core/assets/CCAsset.js:70 |

_native

Serializable url for native asset.

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/assets/CCAsset.js:108 |

_nativeAsset

The underlying native asset of this asset if one is available. This property can be used to access additional details or functionality related to the asset. This property will be initialized by the loader if `_native` is available.

| meta | description |
|------------|--|
| Type | Object |
| Defined in | cocos2d/core/assets/CCAsset.js:116 |

_uuid

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/assets/CCRawAsset.js:46 |

_name

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:76 |

_objFlags

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/platform/CCObject.js:83 |

name

The name of the object.

| meta | description |
|------------|---|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:243 |

Examples

```
obj.name = "New Obj";
```

isValid

Indicates whether the object is not yet destroyed. (It will not be available after being destroyed)
When an object's `destroy` is called, it is actually destroyed after the end of this frame.
So `isValid` will return false from the next frame, while `isValid` in the current frame will still be true.
If you want to determine whether the current frame has called `destroy`, use `cc.isValid(obj, true)`, but this is often caused by a particular logical requirements, which is not normally required.

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:261 |

Examples

```
var node = new cc.Node();
cc.log(node.isValid);      // true
node.destroy();
cc.log(node.isValid);      // true, still valid in this frame
// after a frame...
cc.log(node.isValid);      // false, destroyed in the end of last frame
```

Methods

toString

Returns the asset's url.

The Asset object overrides the `toString()` method of the Object object. For Asset objects, the `toString()` method returns a string representation of the object. JavaScript calls the `toString()` method automatically when an asset is to be represented as a text value or when a texture is referred to in a string concatenation.

| meta | description |
|------------|--|
| Returns | String |
| Defined in | cocos2d/core/assets/CCAsset.js:165 |

serialize

应 AssetDB 要求提供这个方法

| meta | description |
|------------|--|
| Returns | String |
| Defined in | cocos2d/core/assets/CCAsset.js:179 |

createNode

Create a new node using this asset in the scene.

If this type of asset dont have its corresponding node type, this method should be null.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/assets/CCAsset.js:190 |

Parameters

- callback [Function](#)
- error [String](#) null or the error info
- node [Object](#) the created node or null

_setRawAsset

Set native file name for this asset.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/assets/CCAsset.js:205 |

Parameters

- `filename` [String](#)
- `inLibrary` [Boolean](#)

destroy

Destroy this Object, and release all its own references to other objects.

Actual object destruction will delayed until before rendering. From the next frame, this object is not usable any more. You can use `cc.isValid(obj)` to check whether the object is destroyed before accessing it.

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:296 |

Examples

```
obj.destroy();
```

```
_destruct
```

Clear all references in the instance.

NOTE: this method will not clear the getter or setter functions which defined in the instance of CCObject. You can override the `_destruct` method if you need, for example:

```
_destruct: function () { for (var key in this) { if (this.hasOwnProperty(key)) { switch (typeof this[key]) { case 'string': this[key] = ""; break; case 'object': case 'function': this[key] = null; break; } } }}
```

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:430 |

```
_onPreDestroy
```

Called before the object being destroyed.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:463 |

_serialize

The customized serialization for this object. (Editor Only)

| meta | description |
|------------|---|
| Returns | object |
| Defined in | cocos2d/core/platform/CCObject.js:488 |

Parameters

- `exporting Boolean`

_deserialize

Init this object from the custom serialized data.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:498 |

Parameters

- `data Object` the serialized json data
- `ctx _Deserializer`

BlockInputEvents Class

Extends [Component](#)

Module: [cc](#)

This component will block all input events (mouse and touch) within the bounding box of the node, preventing the input from penetrating into the underlying node, typically for the background of the top UI.

This component does not have any API interface and can be added directly to the scene to take effect.

Index

Properties

- `__eventTargets` Array Register all related EventTargets,...
- `node` Node The node this component is attached to.
- `uuid` String The uuid for editor.
- `_enabled` Boolean
- `enabled` Boolean indicates whether this component is enabled or not.
- `enabledInHierarchy` Boolean indicates whether this component is enabled and its node is also active in the hierarchy.
- `_isOnLoadCalled` Number Returns a value which used to indicate the onLoad get called or not.
- `_name` String
- `_objFlags` Number
- `name` String The name of the object.
- `isValid` Boolean Indicates whether the object is not yet destroyed.

Methods

- `update` This is a lifecycle method.
- `lateUpdate` This is a lifecycle method.
- `__preload` __preload is called before every onLoad.
- `onLoad` When attaching to an active node or its node first activated.
- `start` Called before all scripts' update if the Component is enabled the first time.
- `onEnable` This is a lifecycle method.
- `onDisable` This is a lifecycle method.
- `onDestroy` This is a lifecycle method.
- `onFocusInEditor`
- `onLostFocusInEditor`
- `resetInEditor` Called to initialize the component or node's properties when adding the component the first time or when the Reset command is used.
- `addComponent` Adds a component class to the node.
- `getComponent` Returns the component of supplied type if the node has one attached, null if it doesn't....
- `getComponents` Returns all components of supplied Type in the node.
- `getComponentInChildren` Returns the component of supplied type in any of its children using depth first search.
- `getComponentsInChildren` Returns the components of supplied type in self or any of its children using depth first search.
- `_getLocalBounds` If the component's bounding box is different from the node's, you can implement this method to supply
- `onRestore` for undo/redo operation.
- `schedule` Schedules a custom selector....
- `scheduleOnce` Schedules a callback function that runs only once, with a delay of 0 or larger.
- `unschedule` Unschedules a custom callback function.
- `unscheduleAllCallbacks` unschedule all scheduled callback functions: custom callback functions, and the 'update' callback function....
- `destroy` Actual object destruction will delayed until before rendering.
- `_destruct` Clear all references in the instance.
- `_onPreDestroy` Called before the object being destroyed.
- `_serialize` The customized serialization for this object.
- `_deserialize` Init this object from the custom serialized data.

Details

Properties

__eventTargets

Register all related EventTargets, all event callbacks will be removed in _onPreDestroy

| meta | description |
|------------|---|
| Type | Array |
| Defined in | cocos2d/core/components/CCComponent.js:61 |

node

The node this component is attached to. A component is always attached to a node.

| meta | description |
|------------|---|
| Type | Node |
| Defined in | cocos2d/core/components/CCComponent.js:75 |

Examples

```
cc.log(comp.node);
```

uuid

The uuid for editor.

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/components/CCComponent.js:106 |

Examples

```
cc.log(comp.uuid);
```

_enabled

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:147 |

enabled

indicates whether this component is enabled or not.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:154 |

Examples

```
comp.enabled = true;  
cc.log(comp.enabled);
```

enabledInHierarchy

indicates whether this component is enabled and its node is also active in the hierarchy.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:185 |

Examples

```
cc.log(comp.enabledInHierarchy);
```

_isOnLoadCalled

Returns a value which used to indicate the onLoad get called or not.

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/components/CCComponent.js:201 |

Examples

```
cc.log(this.\_isOnLoadCalled > 0);
```

`_name`

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:76 |

`_objFlags`

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/platform/CCObject.js:83 |

`name`

The name of the object.

| meta | description |
|------------|---|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:243 |

Examples

```
obj.name = "New Obj";
```

isValid

Indicates whether the object is not yet destroyed. (It will not be available after being destroyed)
When an object's `destroy` is called, it is actually destroyed after the end of this frame.
So `isValid` will return false from the next frame, while `isValid` in the current frame will still be true.
If you want to determine whether the current frame has called `destroy`, use `cc.isValid(obj, true)`, but this is often caused by a particular logical requirements, which is not normally required.

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:261 |

Examples

```
var node = new cc.Node();
cc.log(node.isValid);      // true
node.destroy();
cc.log(node.isValid);      // true, still valid in this frame
// after a frame...
cc.log(node.isValid);      // false, destroyed in the end of last frame
```

Methods

update

Update is called every frame, if the Component is enabled.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:222 |

Parameters

- `dt` [Number](#) the delta time in seconds it took to complete the last frame

lateUpdate

LateUpdate is called every frame, if the Component is enabled.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:233 |

__preload

__preload is called before every onLoad. It is used to initialize the builtin components internally, to avoid checking whether onLoad is called before every public method calls. This method should be removed if script priority is supported.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:243 |

onLoad

When attaching to an active node or its node first activated. onLoad is always called before any start functions, this allows you to order initialization of scripts.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:254 |

start

Called before all scripts' update if the Component is enabled the first time. Usually used to initialize some logic which need to be called after all components' onload methods called.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:267 |

onEnable

Called when this component becomes enabled and its node is active.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:280 |

onDisable

Called when this component becomes disabled or its node becomes inactive.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:290 |

onDestroy

Called when this component will be destroyed.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:300 |

onFocusInEditor

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:310 |

onLostFocusInEditor

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:315 |

resetInEditor

Called to initialize the component or node's properties when adding the component the first time or when the Reset command is used. This function is only called in editor.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:320 |

addComponent

Adds a component class to the node. You can also add component to node by passing in the name of the script.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:330 |

Parameters

- typeOrClassName [Function](#) | [String](#) the constructor or the class name of the component to add

Examples

```
var sprite = node.addComponent(cc.Sprite);
var test = node.addComponent("Test");
```

getComponent

Returns the component of supplied type if the node has one attached, null if it doesn't. You can also get component in the node by passing in the name of the script.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:348 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
// get sprite component.
var sprite = node.getComponent(cc.Sprite);
// get custom test calss.
var test = node.getComponent("Test");
```

getComponents

Returns all components of supplied Type in the node.

| meta | description |
|------------|--|
| Returns | Component[] |
| Defined in | cocos2d/core/components/CCComponent.js:372 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
var sprites = node.getComponents(cc.Sprite);
var tests = node.getComponents("Test");
```

getComponentInChildren

Returns the component of supplied type in any of its children using depth first search.

| meta | description |
|---------|---------------------------|
| Returns | Component |

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:390 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
var sprite = node.getComponentInChildren(cc.Sprite);
var Test = node.getComponentInChildren("Test");
```

getComponentsInChildren

Returns the components of supplied type in self or any of its children using depth first search.

| meta | description |
|------------|--|
| Returns | Component[] |
| Defined in | cocos2d/core/components/CCComponent.js:408 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
var sprites = node.getComponentsInChildren(cc.Sprite);
var tests = node.getComponentsInChildren("Test");
```

_getLocalBounds

If the component's bounding box is different from the node's, you can implement this method to supply a custom axis aligned bounding box (AABB), so the editor's scene view can perform hit test properly.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:428 |

Parameters

- `out_rect Rect` the Rect to receive the bounding box

onRestore

onRestore is called after the user clicks the Reset item in the Inspector's context menu or performs an undo operation on this component.

If the component contains the "internal state", short for "temporary member variables which not included in its CCClass properties", then you may need to implement this function.

The editor will call the getset accessors of your component to record/restore the component's state for undo/redo operation. However, in extreme cases, it may not work well. Then you should implement this function to manually synchronize your component's "internal states" with its public properties. Once you implement this function, all the getset accessors of your component will not be called when the user performs an undo/redo operation. Which means that only the properties with default value will be recorded or restored by editor.

Similarly, the editor may fail to reset your component correctly in extreme cases. Then if you need to support the reset menu, you should manually synchronize your component's "internal states" with its properties in this function. Once you implement this function, all the getset accessors of your component will not be called during reset operation. Which means that only the properties with default value will be reset by editor.

This function is only called in editor mode.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:441 |

schedule

Schedules a custom selector.

If the selector is already scheduled, then the interval parameter will be updated without scheduling it again.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:535 |

Parameters

- `callback function` The callback function
- `interval Number` Tick interval in seconds. 0 means tick every frame.
- `repeat Number` The selector will be executed (repeat + 1) times, you can use cc.macro.REPEAT_FOREVER for tick infinitely.
- `delay Number` The amount of time that the first tick will wait before execution.

Examples

```
var timeCallback = function (dt) {
    cc.log("time: " + dt);
}
this.schedule(timeCallback, 1);
```

scheduleOnce

Schedules a callback function that runs only once, with a delay of 0 or larger.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:572 |

Parameters

- `callback function` A function wrapped as a selector
- `delay Number` The amount of time that the first tick will wait before execution.

Examples

```
var timeCallback = function (dt) {
    cc.log("time: " + dt);
}
this.scheduleOnce(timeCallback, 2);
```

unschedule

Unschedules a custom callback function.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:589 |

Parameters

- `callback_fn` [function](#) A function wrapped as a selector

Examples

```
this.unschedule(_callback);
```

unscheduleAllCallbacks

unschedule all scheduled callback functions: custom callback functions, and the 'update' callback function.

Actions are not affected by this method.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:605 |

Examples

```
this.unscheduleAllCallbacks();
```

destroy

Destroy this Object, and release all its own references to other objects.

Actual object destruction will delayed until before rendering. From the next frame, this object is not usable any more. You can use `cc.isValid(obj)` to check whether the object is destroyed before accessing it.

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:296 |

Examples

```
obj.destroy();
```

_destruct

Clear all references in the instance.

NOTE: this method will not clear the getter or setter functions which defined in the instance of CCObject. You can override the _destruct method if you need, for example: _destruct: function () { for (var key in this) { if (this.hasOwnProperty(key)) { switch (typeof this[key]) { case 'string': this[key] = ""; break; case 'object': case 'function': this[key] = null; break; } } }

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:430 |

_onPreDestroy

Called before the object being destroyed.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:463 |

_serialize

The customized serialization for this object. (Editor Only)

| meta | description |
|------------|---|
| Returns | object |
| Defined in | cocos2d/core/platform/CCObject.js:488 |

Parameters

- exporting [Boolean](#)

_deserialize

Init this object from the custom serialized data.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:498 |

Parameters

- `data Object` the serialized json data
- `ctx _Deserializer`

BoxCollider Class

Extends `Collider, Collider.Box`(mixin)

Module: `cc`

Box Collider.

Index

Properties

- `offset Vec2` Position offset
- `size Size` Box size
- `tag Integer` Tag.
- `__eventTargets Array` Register all related EventTargets,...
- `node Node` The node this component is attached to.
- `uuid String` The uuid for editor.
- `_enabled Boolean`
- `enabled Boolean` indicates whether this component is enabled or not.
- `enabledInHierarchy Boolean` indicates whether this component is enabled and its node is also active in the hierarchy.
- `_isOnLoadCalled Number` Returns a value which used to indicate the onLoad get called or not.
- `_name String`
- `_objFlags Number`
- `name String` The name of the object.
- `isValid Boolean` Indicates whether the object is not yet destroyed.

Methods

- `update` This is a lifecycle method.
- `lateUpdate` This is a lifecycle method.
- `__preload` __preload is called before every onLoad.
- `onLoad` When attaching to an active node or its node first activated.
- `start` Called before all scripts' update if the Component is enabled the first time.
- `onEnable` This is a lifecycle method.
- `onDisable` This is a lifecycle method.
- `onDestroy` This is a lifecycle method.
- `onFocusInEditor`
- `onLostFocusInEditor`

- `resetInEditor` Called to initialize the component or node's properties when adding the component the first time or when the Reset command is used.
- `addComponent` Adds a component class to the node.
- `getComponent` Returns the component of supplied type if the node has one attached, null if it doesn't....
- `getComponents` Returns all components of supplied Type in the node.
- `getComponentInChildren` Returns the component of supplied type in any of its children using depth first search.
- `getComponentsInChildren` Returns the components of supplied type in self or any of its children using depth first search.
- `_getLocalBounds` If the component's bounding box is different from the node's, you can implement this method to supply
- `onRestore` for undo/redo operation.
- `schedule` Schedules a custom selector....
- `scheduleOnce` Schedules a callback function that runs only once, with a delay of 0 or larger.
- `unschedule` Unschedules a custom callback function.
- `unscheduleAllCallbacks` unschedule all scheduled callback functions: custom callback functions, and the 'update' callback function....
- `destroy` Actual object destruction will delayed until before rendering.
- `_destruct` Clear all references in the instance.
- `_onPreDestroy` Called before the object being destroyed.
- `_serialize` The customized serialization for this object.
- `_deserialize` Init this object from the custom serialized data.

Details

Properties

`offset`

Position offset

| meta | description |
|------------|---|
| Type | Vec2 |
| Defined in | cocos2d/core/collider/CCBoxCollider.js:41 |

`size`

Box size

| meta | description |
|------|----------------------|
| Type | Size |

| meta | description |
|------------|---|
| Defined in | cocos2d/core/collider/CCBoxCollider.js:58 |

tag

Tag. If a node has several collider components, you can judge which type of collider is collided according to the tag.

| meta | description |
|------------|--|
| Type | Integer |
| Defined in | cocos2d/core/collider/CCCollider.js:47 |

__eventTargets

Register all related EventTargets, all event callbacks will be removed in _onPreDestroy

| meta | description |
|------------|---|
| Type | Array |
| Defined in | cocos2d/core/components/CCComponent.js:61 |

node

The node this component is attached to. A component is always attached to a node.

| meta | description |
|------------|---|
| Type | Node |
| Defined in | cocos2d/core/components/CCComponent.js:75 |

Examples

```
cc.log(comp.node);
```

uuid

The uuid for editor.

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/components/CCComponent.js:106 |

Examples

```
cc.log(comp.uuid);
```

_enabled

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:147 |

enabled

indicates whether this component is enabled or not.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:154 |

Examples

```
comp.enabled = true;  
cc.log(comp.enabled);
```

enabledInHierarchy

indicates whether this component is enabled and its node is also active in the hierarchy.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:185 |

Examples

```
cc.log(comp.enabledInHierarchy);
```

_isOnLoadCalled

Returns a value which used to indicate the onLoad get called or not.

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/components/CCComponent.js:201 |

Examples

```
cc.log(this._isOnLoadCalled > 0);
```

_name

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:76 |

_objFlags

| meta | description |
|------|------------------------|
| Type | Number |

| meta | description |
|------------|--|
| Defined in | cocos2d/core/platform/CCObject.js:83 |

name

The name of the object.

| meta | description |
|------------|---|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:243 |

Examples

```
obj.name = "New Obj";
```

isValid

Indicates whether the object is not yet destroyed. (It will not be available after being destroyed)
When an object's `destroy` is called, it is actually destroyed after the end of this frame.
So `isValid` will return false from the next frame, while `isValid` in the current frame will still be true.
If you want to determine whether the current frame has called `destroy`, use `cc.isValid(obj, true)`, but this is often caused by a particular logical requirements, which is not normally required.

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:261 |

Examples

```
var node = new cc.Node();
cc.log(node.isValid);    // true
node.destroy();
cc.log(node.isValid);    // true, still valid in this frame
// after a frame...
cc.log(node.isValid);    // false, destroyed in the end of last frame
```

Methods

update

Update is called every frame, if the Component is enabled.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:222 |

Parameters

- `dt` `Number` the delta time in seconds it took to complete the last frame

lateUpdate

LateUpdate is called every frame, if the Component is enabled.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:233 |

__preload

__preload is called before every onLoad. It is used to initialize the builtin components internally, to avoid checking whether onLoad is called before every public method calls. This method should be removed if script priority is supported.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:243 |

onLoad

When attaching to an active node or its node first activated. onLoad is always called before any start functions, this allows you to order initialization of scripts.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:254 |

start

Called before all scripts' update if the Component is enabled the first time. Usually used to initialize some logic which need to be called after all components' `onload` methods called. This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:267 |

onEnable

Called when this component becomes enabled and its node is active. This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:280 |

onDisable

Called when this component becomes disabled or its node becomes inactive. This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:290 |

onDestroy

Called when this component will be destroyed. This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|-------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:300 |

onFocusInEditor

| meta | description |
|-------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:310 |

onLostFocusInEditor

| meta | description |
|-------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:315 |

resetInEditor

Called to initialize the component or node's properties when adding the component the first time or when the Reset command is used. This function is only called in editor.

| meta | description |
|-------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:320 |

addComponent

Adds a component class to the node. You can also add component to node by passing in the name of the script.

| meta | description |
|-------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:330 |

Parameters

- `typeOrClassName` [Function](#) | [String](#) the constructor or the class name of the component to add

Examples

```
var sprite = node.addComponent(cc.Sprite);
var test = node.addComponent("Test");
```

getComponent

Returns the component of supplied type if the node has one attached, null if it doesn't. You can also get component in the node by passing in the name of the script.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:348 |

Parameters

- `typeOrClassName` [Function](#) | [String](#)

Examples

```
// get sprite component.
var sprite = node.getComponent(cc.Sprite);
// get custom test calss.
var test = node.getComponent("Test");
```

getComponents

Returns all components of supplied Type in the node.

| meta | description |
|------------|--|
| Returns | Component[] |
| Defined in | cocos2d/core/components/CCComponent.js:372 |

Parameters

- `typeOrClassName` [Function](#) | [String](#)

Examples

```
var sprites = node.getComponents(cc.Sprite);
var tests = node.getComponents("Test");
```

getComponentInChildren

Returns the component of supplied type in any of its children using depth first search.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:390 |

Parameters

- `typeOrClassName` [Function](#) | [String](#)

Examples

```
var sprite = node.getComponentInChildren(cc.Sprite);
var Test = node.getComponentInChildren("Test");
```

getComponentsInChildren

Returns the components of supplied type in self or any of its children using depth first search.

| meta | description |
|------------|--|
| Returns | Component[] |
| Defined in | cocos2d/core/components/CCComponent.js:408 |

Parameters

- `typeOrClassName` [Function](#) | [String](#)

Examples

```
var sprites = node.getComponentsInChildren(cc.Sprite);
var tests = node.getComponentsInChildren("Test");
```

_getLocalBounds

If the component's bounding box is different from the node's, you can implement this method to supply a custom axis aligned bounding box (AABB), so the editor's scene view can perform hit test properly.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:428 |

Parameters

- `out_rect` [Rect](#) the Rect to receive the bounding box

onRestore

onRestore is called after the user clicks the Reset item in the Inspector's context menu or performs an undo operation on this component.

If the component contains the "internal state", short for "temporary member variables which not included in its CCClass properties", then you may need to implement this function.

The editor will call the getset accessors of your component to record/restore the component's state for undo/redo operation. However, in extreme cases, it may not work well. Then you should implement this function to manually synchronize your component's "internal states" with its public properties. Once you implement this function, all the getset accessors of your component will not be called when the user performs an undo/redo operation. Which means that only the properties with default value will be recorded or restored by editor.

Similarly, the editor may fail to reset your component correctly in extreme cases. Then if you need to support the reset menu, you should manually synchronize your component's "internal states" with its properties in this function. Once you implement this function, all the getset accessors of your component will not be called during reset operation. Which means that only the properties with default value will be reset by editor.

This function is only called in editor mode.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:441 |

schedule

Schedules a custom selector.

If the selector is already scheduled, then the interval parameter will be updated without scheduling it again.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:535 |

Parameters

- `callback` [function](#) The callback function
- `interval` [Number](#) Tick interval in seconds. 0 means tick every frame.
- `repeat` [Number](#) The selector will be executed (repeat + 1) times, you can use cc.macro.REPEAT_FOREVER for tick infinitely.
- `delay` [Number](#) The amount of time that the first tick will wait before execution.

Examples

```
var timeCallback = function (dt) {
    cc.log("time: " + dt);
}
this.schedule(timeCallback, 1);
```

scheduleOnce

Schedules a callback function that runs only once, with a delay of 0 or larger.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:572 |

Parameters

- `callback` [function](#) A function wrapped as a selector
- `delay` [Number](#) The amount of time that the first tick will wait before execution.

Examples

```
var timeCallback = function (dt) {  
    cc.log("time: " + dt);  
}  
this.scheduleOnce(timeCallback, 2);
```

unschedule

Unschedules a custom callback function.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:589 |

Parameters

- `callback_fn` [function](#) A function wrapped as a selector

Examples

```
this.unschedule(_callback);
```

unscheduleAllCallbacks

unschedule all scheduled callback functions: custom callback functions, and the 'update' callback function.

Actions are not affected by this method.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:605 |

Examples

```
this.unscheduleAllCallbacks();
```

destroy

Destroy this Object, and release all its own references to other objects.

Actual object destruction will delayed until before rendering. From the next frame, this object is not usable any more. You can use `cc.isValid(obj)` to check whether the object is destroyed before accessing it.

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:296 |

Examples

```
obj.destroy();
```

_destruct

Clear all references in the instance.

NOTE: this method will not clear the getter or setter functions which defined in the instance of CCObject. You can override the _destruct method if you need, for example: _destruct: function () { for (var key in this) { if (this.hasOwnProperty(key)) { switch (typeof this[key]) { case 'string': this[key] = ''; break; case 'object': case 'function': this[key] = null; break; } } }

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:430 |

_onPreDestroy

Called before the object being destroyed.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:463 |

_serialize

The customized serialization for this object. (Editor Only)

| meta | description |
|------------|---|
| Returns | object |
| Defined in | cocos2d/core/platform/CCObject.js:488 |

Parameters

- `exporting Boolean`

`_deserialize`

Init this object from the custom serialized data.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:498 |

Parameters

- `data Object` the serialized json data
- `ctx _Deserializer`

Button Class

Extends [Component](#)

Module: [cc](#)

Button has 4 Transition types

When Button state changed:

If Transition type is Button.Transition.NONE, Button will do nothing

If Transition type is Button.Transition.COLOR, Button will change target's color

If Transition type is Button.Transition.SPRITE, Button will change target Sprite's sprite

If Transition type is Button.Transition.SCALE, Button will change target node's scale

Button will trigger 5 events:

Button.EVENT_TOUCH_DOWN

Button.EVENT_TOUCH_UP

Button.EVENT_HOVER_IN

Button.EVENT_HOVER_MOVE

Button.EVENT_HOVER_OUT

Index

Properties

- `interactable Boolean` Whether the Button is disabled.

- `enableAutoGrayEffect` Boolean When this flag is true, Button target sprite will turn gray when interactable is false.
- `transition` `Button.Transition` Transition type
- `normalColor` `Color` Normal state color.
- `pressedColor` `Color` Pressed state color
- `hoverColor` `Color` Hover state color
- `disabledColor` `Color` Disabled state color
- `duration` `Number` Color and Scale transition duration
- `zoomScale` `Number` When user press the button, the button will zoom to a scale.
- `normalSprite` `SpriteFrame` Normal state sprite
- `pressedSprite` `SpriteFrame` Pressed state sprite
- `hoverSprite` `SpriteFrame` Hover state sprite
- `disabledSprite` `SpriteFrame` Disabled state sprite
- `target` `Node` Transition target.
- `clickEvents` `Component.EventHandler[]` If Button is clicked, it will trigger event's handler
- `__eventTargets` `Array` Register all related EventTargets,...
- `node` `Node` The node this component is attached to.
- `uuid` `String` The uuid for editor.
- `_enabled` `Boolean`
- `enabled` `Boolean` indicates whether this component is enabled or not.
- `enabledInHierarchy` `Boolean` indicates whether this component is enabled and its node is also active in the hierarchy.
- `_isOnLoadCalled` `Number` Returns a value which used to indicate the onLoad get called or not.
- `_name` `String`
- `_objFlags` `Number`
- `name` `String` The name of the object.
- `isValid` `Boolean` Indicates whether the object is not yet destroyed.

Methods

- `update` This is a lifecycle method.
- `lateUpdate` This is a lifecycle method.
- `__preload` `__preload` is called before every onLoad.
- `onLoad` When attaching to an active node or its node first activated.
- `start` Called before all scripts' update if the Component is enabled the first time.
- `onEnable` This is a lifecycle method.
- `onDisable` This is a lifecycle method.
- `onDestroy` This is a lifecycle method.
- `onFocusInEditor`
- `onLostFocusInEditor`
- `resetInEditor` Called to initialize the component or node's properties when adding the component the first time or when the Reset command is used.
- `addComponent` Adds a component class to the node.
- `getComponent` Returns the component of supplied type if the node has one attached, null if it doesn't....
- `getComponents` Returns all components of supplied Type in the node.
- `getComponentInChildren` Returns the component of supplied type in any of its children using depth first search.
- `getComponentsInChildren` Returns the components of supplied type in self or any of its children using depth first search.

- `_getLocalBounds` If the component's bounding box is different from the node's, you can implement this method to supply `onRestore` for undo/redo operation.
- `schedule` Schedules a custom selector....
- `scheduleOnce` Schedules a callback function that runs only once, with a delay of 0 or larger.
- `unschedule` Unschedules a custom callback function.
- `unscheduleAllCallbacks` unschedule all scheduled callback functions: custom callback functions, and the 'update' callback function....
- `destroy` Actual object destruction will delayed until before rendering.
- `_destruct` Clear all references in the instance.
- `_onPreDestroy` Called before the object being destroyed.
- `_serialize` The customized serialization for this object.
- `_deserialize` Init this object from the custom serialized data.

Events

- `click` Note: This event is emitted from the node to which the component belongs.

Details

Properties

interactable

Whether the Button is disabled. If true, the Button will trigger event and do transition.

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCButton.js:152 |

enableAutoGrayEffect

When this flag is true, Button target sprite will turn gray when interactable is false.

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCButton.js:190 |

transition

Transition type

| meta | description |
|------------|---|
| Type | Button.Transition |
| Defined in | cocos2d/core/components/CCButton.js:203 |

normalColor

Normal state color.

| meta | description |
|------------|---|
| Type | Color |
| Defined in | cocos2d/core/components/CCButton.js:218 |

pressedColor

Pressed state color

| meta | description |
|------------|---|
| Type | Color |
| Defined in | cocos2d/core/components/CCButton.js:232 |

hoverColor

Hover state color

| meta | description |
|------|-----------------------|
| Type | Color |

| meta | description |
|-------------|---|
| Defined in | cocos2d/core/components/CCButton.js:243 |

disabledColor

Disabled state color

| meta | description |
|-------------|---|
| Type | Color |
| Defined in | cocos2d/core/components/CCButton.js:254 |

duration

Color and Scale transition duration

| meta | description |
|-------------|---|
| Type | Number |
| Defined in | cocos2d/core/components/CCButton.js:268 |

zoomScale

When user press the button, the button will zoom to a scale. The final scale of the button equals (button original scale * zoomScale)

| meta | description |
|-------------|---|
| Type | Number |
| Defined in | cocos2d/core/components/CCButton.js:279 |

normalSprite

Normal state sprite

| meta | description |
|------------|---|
| Type | SpriteFrame |
| Defined in | cocos2d/core/components/CCButton.js:291 |

pressedSprite

Pressed state sprite

| meta | description |
|------------|---|
| Type | SpriteFrame |
| Defined in | cocos2d/core/components/CCButton.js:306 |

hoverSprite

Hover state sprite

| meta | description |
|------------|---|
| Type | SpriteFrame |
| Defined in | cocos2d/core/components/CCButton.js:322 |

disabledSprite

Disabled state sprite

| meta | description |
|------------|---|
| Type | SpriteFrame |
| Defined in | cocos2d/core/components/CCButton.js:338 |

target

Transition target. When Button state changed: If Transition type is Button.Transition.NONE, Button will do nothing If Transition type is Button.Transition.COLOR, Button will change target's color If Transition type is Button.Transition.SPRITE, Button will change target Sprite's sprite

| meta | description |
|------------|---|
| Type | Node |
| Defined in | cocos2d/core/components/CCButton.js:353 |

clickEvents

If Button is clicked, it will trigger event's handler

| meta | description |
|------------|---|
| Type | Component.EventHandler[] |
| Defined in | cocos2d/core/components/CCButton.js:377 |

__eventTargets

Register all related EventTargets, all event callbacks will be removed in `_onPreDestroy`

| meta | description |
|------------|---|
| Type | Array |
| Defined in | cocos2d/core/components/CCComponent.js:61 |

node

The node this component is attached to. A component is always attached to a node.

| meta | description |
|------------|---|
| Type | Node |
| Defined in | cocos2d/core/components/CCComponent.js:75 |

Examples

```
cc.log(comp.node);
```

uuid

The uuid for editor.

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/components/CCComponent.js:106 |

Examples

```
cc.log(comp.uuid);
```

_enabled

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:147 |

enabled

indicates whether this component is enabled or not.

| meta | description |
|------|-------------------------|
| Type | Boolean |

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:154 |

Examples

```
comp.enabled = true;
cc.log(comp.enabled);
```

enabledInHierarchy

indicates whether this component is enabled and its node is also active in the hierarchy.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:185 |

Examples

```
cc.log(comp.enabledInHierarchy);
```

_isOnLoadCalled

Returns a value which used to indicate the onLoad get called or not.

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/components/CCComponent.js:201 |

Examples

```
cc.log(this._isOnLoadCalled > 0);
```

_name

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:76 |

_objFlags

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/platform/CCObject.js:83 |

name

The name of the object.

| meta | description |
|------------|---|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:243 |

Examples

```
obj.name = "New Obj";
```

isValid

Indicates whether the object is not yet destroyed. (It will not be available after being destroyed)

When an object's `destroy` is called, it is actually destroyed after the end of this frame. So `isValid` will return false from the next frame, while `isValid` in the current frame will still be true. If you want to determine whether the current frame has called `destroy`, use `cc.isValid(obj, true)`, but this is often caused by a particular logical requirements, which is not normally required.

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:261 |

Examples

```
var node = new cc.Node();
cc.log(node.isValid);    // true
node.destroy();
cc.log(node.isValid);    // true, still valid in this frame
// after a frame...
cc.log(node.isValid);    // false, destroyed in the end of last frame
```

Methods

update

Update is called every frame, if the Component is enabled.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:222 |

Parameters

- dt [Number](#) the delta time in seconds it took to complete the last frame

lateUpdate

LateUpdate is called every frame, if the Component is enabled.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:233 |

__preload

__preload is called before every `onLoad`. It is used to initialize the builtin components internally, to avoid checking whether `onLoad` is called before every public method calls. This method should be removed if script priority is supported.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:243 |

onLoad

When attaching to an active node or its node first activated. `onLoad` is always called before any start functions, this allows you to order initialization of scripts. This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:254 |

start

Called before all scripts' update if the Component is enabled the first time. Usually used to initialize some logic which need to be called after all components' `onload` methods called.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:267 |

onEnable

Called when this component becomes enabled and its node is active.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:280 |

onDisable

Called when this component becomes disabled or its node becomes inactive.
This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:290 |

onDestroy

Called when this component will be destroyed.
This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:300 |

onFocusInEditor

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:310 |

onLostFocusInEditor

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:315 |

resetInEditor

Called to initialize the component or node's properties when adding the component the first time or when the Reset command is used. This function is only called in editor.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:320 |

addComponent

Adds a component class to the node. You can also add component to node by passing in the name of the script.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:330 |

Parameters

- typeOrClassName [Function](#) | [String](#) the constructor or the class name of the component to add

Examples

```
var sprite = node.addComponent(cc.Sprite);
var test = node.addComponent("Test");
```

getComponent

Returns the component of supplied type if the node has one attached, null if it doesn't.

You can also get component in the node by passing in the name of the script.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:348 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
// get sprite component.  
var sprite = node.getComponent(cc.Sprite);  
// get custom test calss.  
var test = node.getComponent("Test");
```

getComponents

Returns all components of supplied Type in the node.

| meta | description |
|------------|--|
| Returns | Component[] |
| Defined in | cocos2d/core/components/CCComponent.js:372 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
var sprites = node.getComponents(cc.Sprite);  
var tests = node.getComponents("Test");
```

getComponentInChildren

Returns the component of supplied type in any of its children using depth first search.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:390 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
var sprite = node.getComponentInChildren(cc.Sprite);  
var Test = node.getComponentInChildren("Test");
```

getComponentsInChildren

Returns the components of supplied type in self or any of its children using depth first search.

| meta | description |
|------------|--|
| Returns | Component[] |
| Defined in | cocos2d/core/components/CCComponent.js:408 |

Parameters

- `typeOrClassName` [Function](#) | [String](#)

Examples

```
var sprites = node.getComponentsInChildren(cc.Sprite);
var tests = node.getComponentsInChildren("Test");
```

_getLocalBounds

If the component's bounding box is different from the node's, you can implement this method to supply a custom axis aligned bounding box (AABB), so the editor's scene view can perform hit test properly.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:428 |

Parameters

- `out_rect` [Rect](#) the Rect to receive the bounding box

onRestore

onRestore is called after the user clicks the Reset item in the Inspector's context menu or performs an undo operation on this component.

If the component contains the "internal state", short for "temporary member variables which not included in its CCClass properties", then you may need to implement this function.

The editor will call the getset accessors of your component to record/restore the component's state for undo/redo operation. However, in extreme cases, it may not work well. Then you should implement this function to manually synchronize your component's "internal states" with its public properties. Once you implement this function, all the getset accessors of your component will not be called when the user performs an undo/redo operation. Which means that only the properties with default value will be recorded or restored by editor.

Similarly, the editor may fail to reset your component correctly in extreme cases. Then if you need to support the reset menu, you should manually synchronize your component's "internal states" with its properties in this function. Once you implement this function, all the getset accessors of your component will not be called during reset operation. Which means that only the properties with default value will be reset by editor.

This function is only called in editor mode.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:441 |

schedule

Schedules a custom selector. If the selector is already scheduled, then the interval parameter will be updated without scheduling it again.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:535 |

Parameters

- `callback` [function](#) The callback function
- `interval` [Number](#) Tick interval in seconds. 0 means tick every frame.
- `repeat` [Number](#) The selector will be executed (repeat + 1) times, you can use `cc.macro.REPEAT_FOREVER` for tick infinitely.

- `delay` [Number](#) The amount of time that the first tick will wait before execution.

Examples

```
var timeCallback = function (dt) {
    cc.log("time: " + dt);
}
this.schedule(timeCallback, 1);
```

scheduleOnce

Schedules a callback function that runs only once, with a delay of 0 or larger.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:572 |

Parameters

- `callback` [function](#) A function wrapped as a selector
- `delay` [Number](#) The amount of time that the first tick will wait before execution.

Examples

```
var timeCallback = function (dt) {
    cc.log("time: " + dt);
}
this.scheduleOnce(timeCallback, 2);
```

unschedule

Unschedules a custom callback function.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:589 |

Parameters

- `callback_fn` [function](#) A function wrapped as a selector

Examples

```
this.unschedule(_callback);
```

unscheduleAllCallbacks

unschedule all scheduled callback functions: custom callback functions, and the 'update' callback function.

Actions are not affected by this method.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:605 |

Examples

```
this.unscheduleAllCallbacks();
```

destroy

Destroy this Object, and release all its own references to other objects.

Actual object destruction will delayed until before rendering. From the next frame, this object is not usable any more. You can use cc.isValid(obj) to check whether the object is destroyed before accessing it.

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:296 |

Examples

```
obj.destroy();
```

_destruct

Clear all references in the instance.

NOTE: this method will not clear the getter or setter functions which defined in the instance of CCObject. You can override the _destruct method if you need, for example:

```
_destruct: function () { for (var key in this) { if (this.hasOwnProperty(key)) { switch (typeof this[key]) { case 'string': this[key] = ""; break; case 'object': case 'function': this[key] = null; break; } } }
```

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:430 |

_onPreDestroy

Called before the object being destroyed.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:463 |

_serialize

The customized serialization for this object. (Editor Only)

| meta | description |
|------------|---|
| Returns | object |
| Defined in | cocos2d/core/platform/CCObject.js:488 |

Parameters

- exporting [Boolean](#)

_deserialize

Init this object from the custom serialized data.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:498 |

Parameters

- data [Object](#) the serialized json data
- ctx [_Deserializer](#)

Events

click Event

Module: [cc](#)

Note: This event is emitted from the node to which the component belongs.

CallbacksInvoker Class

Extends [_CallbacksHandler](#)

Module: [decorator](#) Parent Module: [cc](#)

The callbacks invoker to handle and invoke callbacks by key.

Index

Methods

- [invoke](#)
- [add](#)
- [hasEventListener](#) Check if the specified key has any registered callback.
- [removeAll](#) Removes all callbacks registered in a certain event type or all callbacks registered with a certain target
- [remove](#)

Details

Methods

invoke

| meta | description |
|------------|--|
| Defined in | cocos2d/core/platform/callbacks-invoker.js:236 |

Parameters

- key [String](#)
- p1 Any
- p2 Any
- p3 Any
- p4 Any
- p5 Any

add

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/callbacks-invoker.js:97 |

Parameters

- key [String](#)
- callback [Function](#)
- target [Object](#) can be null

hasEventListener

Check if the specified key has any registered callback. If a callback is also specified, it will only return true if the callback is registered.

| meta | description |
|------------|--|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/callbacks-invoker.js:112 |

Parameters

- key [String](#)
- callback [Function](#)
- target [Object](#)

removeAll

Removes all callbacks registered in a certain event type or all callbacks registered with a certain target

| meta | description |
|------------|--|
| Defined in | cocos2d/core/platform/callbacks-invoker.js:154 |

Parameters

- keyOrTarget [String](#) | [Object](#) The event key to be removed or the target to be removed

remove

| meta | description |
|------------|--|
| Defined in | cocos2d/core/platform/callbacks-invoker.js:192 |

Parameters

- `key` [String](#)
- `callback` [Function](#)
- `target` [Object](#)

Camera Class

Extends [Component](#)

Module: [cc](#)

Camera is usefull when making reel game or other games which need scroll screen. Using camera will be more efficient than moving node to scroll screen. Camera

Index

Properties

- `zoomRatio` Number The camera zoom ratio.
- `cullingMask` Number This is used to render parts of the scene selectively.
- `clearFlags` Camera.ClearFlags Determining what to clear when camera rendering.
- `backgroundColor` Color The color with which the screen will be cleared.
- `depth` Number Camera's depth in the camera rendering order.
- `targetTexture` RenderTexture Destination render texture.
- `main` Camera The first enabled camera.
- `cameras` [Camera] All enabled cameras.
- `__eventTargets` Array Register all related EventTargets,...
- `node` Node The node this component is attached to.
- `uuid` String The uuid for editor.
- `_enabled` Boolean
- `enabled` Boolean indicates whether this component is enabled or not.
- `enabledInHierarchy` Boolean indicates whether this component is enabled and its node is also active in the hierarchy.
- `_isOnLoadCalled` Number Returns a value which used to indicate the onLoad get called or not.
- `_name` String
- `_objFlags` Number
- `name` String The name of the object.
- `isValid` Boolean Indicates whether the object is not yet destroyed.

Methods

- `findCamera` Get the first camera which the node belong to.
- `getNodeToCameraTransform` Returns the matrix that transform the node's (local) space coordinates into the camera's space coordinates.
- `getCameraToWorldPoint` Conver a camera coordinates point to world coordinates.
- `getWorldToCameraPoint` Conver a world coordinates point to camera coordinates.
- `getCameraToWorldMatrix` Get the camera to world matrix
- `getWorldToCameraMatrix` Get the world to camera matrix
- `containsNode` Check whether the node is in the camera.
- `render` Render the camera manually.
- `update` This is a lifecycle method.
- `lateUpdate` This is a lifecycle method.
- `__preload` `__preload` is called before every `onLoad`.
- `onLoad` When attaching to an active node or its node first activated.
- `start` Called before all scripts' update if the Component is enabled the first time.
- `onEnable` This is a lifecycle method.
- `onDisable` This is a lifecycle method.
- `onDestroy` This is a lifecycle method.
- `onFocusInEditor`
- `onLostFocusInEditor`
- `resetInEditor` Called to initialize the component or node's properties when adding the component the first time or when the Reset command is used.
- `addComponent` Adds a component class to the node.
- `getComponent` Returns the component of supplied type if the node has one attached, null if it doesn't....
- `getComponents` Returns all components of supplied Type in the node.
- `getComponentInChildren` Returns the component of supplied type in any of its children using depth first search.
- `getComponentsInChildren` Returns the components of supplied type in self or any of its children using depth first search.
- `_getLocalBounds` If the component's bounding box is different from the node's, you can implement this method to supply
- `onRestore` for undo/redo operation.
- `schedule` Schedules a custom selector....
- `scheduleOnce` Schedules a callback function that runs only once, with a delay of 0 or larger.
- `unschedule` Unschedules a custom callback function.
- `unscheduleAllCallbacks` unschedule all scheduled callback functions: custom callback functions, and the 'update' callback function....
- `destroy` Actual object destruction will delayed until before rendering.
- `_destruct` Clear all references in the instance.
- `_onPreDestroy` Called before the object being destroyed.
- `_serialize` The customized serialization for this object.
- `_deserialize` Init this object from the custom serialized data.

Details

Properties

zoomRatio

The camera zoom ratio.

| meta | description |
|------------|---|
| Type | Number |
| Defined in | cocos2d/core/camera/CCCamera.js:124 |

cullingMask

This is used to render parts of the scene selectively.

| meta | description |
|------------|---|
| Type | Number |
| Defined in | cocos2d/core/camera/CCCamera.js:141 |

clearFlags

Determining what to clear when camera rendering.

| meta | description |
|------------|---|
| Type | Camera.ClearFlags |
| Defined in | cocos2d/core/camera/CCCamera.js:158 |

backgroundColor

The color with which the screen will be cleared.

| meta | description |
|------|-----------------------|
| Type | Color |

| meta | description |
|------------|---|
| Defined in | cocos2d/core/camera/CCCamera.js:177 |

depth

Camera's depth in the camera rendering order.

| meta | description |
|------------|---|
| Type | Number |
| Defined in | cocos2d/core/camera/CCCamera.js:194 |

targetTexture

Destination render texture. Usually cameras render directly to screen, but for some effects it is useful to make a camera render into a texture.

| meta | description |
|------------|---|
| Type | RenderTarget |
| Defined in | cocos2d/core/camera/CCCamera.js:213 |

main

The first enabled camera.

| meta | description |
|------------|---|
| Type | Camera |
| Defined in | cocos2d/core/camera/CCCamera.js:234 |

cameras

All enabled cameras.

| meta | description |
|------------|---|
| Type | [Camera] |
| Defined in | cocos2d/core/camera/CCCamera.js:244 |

__eventTargets

Register all related EventTargets, all event callbacks will be removed in `_onPreDestroy`

| meta | description |
|------------|---|
| Type | Array |
| Defined in | cocos2d/core/components/CCComponent.js:61 |

node

The node this component is attached to. A component is always attached to a node.

| meta | description |
|------------|---|
| Type | Node |
| Defined in | cocos2d/core/components/CCComponent.js:75 |

Examples

```
cc.log(comp.node);
```

uuid

The uuid for editor.

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/components/CCComponent.js:106 |

Examples

```
cc.log(comp.uuid);
```

_enabled

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:147 |

enabled

indicates whether this component is enabled or not.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:154 |

Examples

```
comp.enabled = true;  
cc.log(comp.enabled);
```

enabledInHierarchy

indicates whether this component is enabled and its node is also active in the hierarchy.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:185 |

Examples

```
cc.log(comp.enabledInHierarchy);
```

_isOnLoadCalled

Returns a value which used to indicate the onLoad get called or not.

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/components/CCComponent.js:201 |

Examples

```
cc.log(this._isOnLoadCalled > 0);
```

_name

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:76 |

_objFlags

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/platform/CCObject.js:83 |

name

The name of the object.

| meta | description |
|------|------------------------|
| Type | String |

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:243 |

Examples

```
obj.name = "New Obj";
```

isValid

Indicates whether the object is not yet destroyed. (It will not be available after being destroyed)
When an object's `destroy` is called, it is actually destroyed after the end of this frame.
So `isValid` will return false from the next frame, while `isValid` in the current frame will still be true.
If you want to determine whether the current frame has called `destroy`, use `cc.isValid(obj, true)`, but this is often caused by a particular logical requirements, which is not normally required.

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:261 |

Examples

```
var node = new cc.Node();
cc.log(node.isValid);      // true
node.destroy();
cc.log(node.isValid);      // true, still valid in this frame
// after a frame...
cc.log(node.isValid);      // false, destroyed in the end of last frame
```

Methods

findCamera

Get the first camera which the node belong to.

| meta | description |
|------------|---|
| Returns | Camera |
| Defined in | cocos2d/core/camera/CCCamera.js:256 |

Parameters

- node [Node](#)

getNodeToCameraTransform

Returns the matrix that transform the node's (local) space coordinates into the camera's space coordinates.

| meta | description |
|------------|---|
| Returns | AffineTransform |
| Defined in | cocos2d/core/camera/CCCamera.js:376 |

Parameters

- node [Node](#) the node which should transform

getCameraToWorldPoint

Conver a camera coordinates point to world coordinates.

| meta | description |
|------------|---|
| Returns | Vec2 |
| Defined in | cocos2d/core/camera/CCCamera.js:396 |

Parameters

- point [Vec2](#) the point which should transform
- out [Vec2](#) the point to receive the result

getWorldToCameraPoint

Conver a world coordinates point to camera coordinates.

| meta | description |
|---------|----------------------|
| Returns | Vec2 |

| meta | description |
|------------|---|
| Defined in | cocos2d/core/camera/CCCamera.js:413 |

Parameters

- `point` [Vec2](#)
- `out` [Vec2](#) the point to receive the result

`getCameraToWorldMatrix`

Get the camera to world matrix

| meta | description |
|------------|---|
| Returns | Mat4 |
| Defined in | cocos2d/core/camera/CCCamera.js:430 |

Parameters

- `out` Mat4 the matrix to receive the result

`getWorldToCameraMatrix`

Get the world to camera matrix

| meta | description |
|------------|---|
| Returns | Mat4 |
| Defined in | cocos2d/core/camera/CCCamera.js:446 |

Parameters

- `out` Mat4 the matrix to receive the result

`containsNode`

Check whether the node is in the camera.

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/camera/CCCamera.js:477 |

Parameters

- node [Node](#) the node which need to check

render

Render the camera manually.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/camera/CCCamera.js:490 |

Parameters

- root [Node](#)

update

Update is called every frame, if the Component is enabled.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:222 |

Parameters

- dt [Number](#) the delta time in seconds it took to complete the last frame

lateUpdate

LateUpdate is called every frame, if the Component is enabled.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:233 |

__preload

`__preload` is called before every `onLoad`. It is used to initialize the builtin components internally, to avoid checking whether `onLoad` is called before every public method calls. This method should be removed if script priority is supported.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:243 |

onLoad

When attaching to an active node or its node first activated. `onLoad` is always called before any start functions, this allows you to order initialization of scripts.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:254 |

start

Called before all scripts' update if the Component is enabled the first time. Usually used to initialize some logic which need to be called after all components' `onload` methods called. This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:267 |

onEnable

Called when this component becomes enabled and its node is active.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|-------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:280 |

onDisable

Called when this component becomes disabled or its node becomes inactive.
This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|-------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:290 |

onDestroy

Called when this component will be destroyed.
This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|-------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:300 |

onFocusInEditor

| meta | description |
|-------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:310 |

onLostFocusInEditor

| meta | description |
|-------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:315 |

resetInEditor

Called to initialize the component or node's properties when adding the component the first time or when the Reset command is used. This function is only called in editor.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:320 |

addComponent

Adds a component class to the node. You can also add component to node by passing in the name of the script.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:330 |

Parameters

- typeOrClassName [Function](#) | [String](#) the constructor or the class name of the component to add

Examples

```
var sprite = node.addComponent(cc.Sprite);
var test = node.addComponent("Test");
```

getComponent

Returns the component of supplied type if the node has one attached, null if it doesn't. You can also get component in the node by passing in the name of the script.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:348 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
// get sprite component.  
var sprite = node.getComponent(cc.Sprite);  
// get custom test calss.  
var test = node.getComponent("Test");
```

getComponents

Returns all components of supplied Type in the node.

| meta | description |
|------------|--|
| Returns | Component[] |
| Defined in | cocos2d/core/components/CCComponent.js:372 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
var sprites = node.getComponents(cc.Sprite);  
var tests = node.getComponents("Test");
```

getComponentInChildren

Returns the component of supplied type in any of its children using depth first search.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:390 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
var sprite = node.getComponentInChildren(cc.Sprite);
var Test = node.getComponentInChildren("Test");
```

getComponentsInChildren

Returns the components of supplied type in self or any of its children using depth first search.

| meta | description |
|------------|--|
| Returns | Component[] |
| Defined in | cocos2d/core/components/CCComponent.js:408 |

Parameters

- `typeOrClassName` [Function](#) | [String](#)

Examples

```
var sprites = node.getComponentsInChildren(cc.Sprite);
var tests = node.getComponentsInChildren("Test");
```

_getLocalBounds

If the component's bounding box is different from the node's, you can implement this method to supply a custom axis aligned bounding box (AABB), so the editor's scene view can perform hit test properly.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:428 |

Parameters

- `out_rect` [Rect](#) the Rect to receive the bounding box

onRestore

onRestore is called after the user clicks the Reset item in the Inspector's context menu or performs an undo operation on this component.

If the component contains the "internal state", short for "temporary member variables which not included in its CCClass properties", then you may need to implement this function.

The editor will call the getset accessors of your component to record/restore the component's state for undo/redo operation. However, in extreme cases, it may not work well. Then you should implement this function to manually synchronize your component's "internal states" with its public properties. Once you implement this function, all the getset accessors of your component will not be called when the user performs an undo/redo operation. Which means that only the properties with default value will be recorded or restored by editor.

Similarly, the editor may fail to reset your component correctly in extreme cases. Then if you need to support the reset menu, you should manually synchronize your component's "internal states" with its properties in this function. Once you implement this function, all the getset accessors of your component will not be called during reset operation. Which means that only the properties with default value will be reset by editor.

This function is only called in editor mode.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:441 |

schedule

Schedules a custom selector.

If the selector is already scheduled, then the interval parameter will be updated without scheduling it again.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:535 |

Parameters

- `callback function` The callback function
- `interval Number` Tick interval in seconds. 0 means tick every frame.
- `repeat Number` The selector will be executed (repeat + 1) times, you can use `cc.macro.REPEAT_FOREVER` for tick infinitely.
- `delay Number` The amount of time that the first tick will wait before execution.

Examples

```
var timeCallback = function (dt) {
    cc.log("time: " + dt);
}
this.schedule(timeCallback, 1);
```

scheduleOnce

Schedules a callback function that runs only once, with a delay of 0 or larger.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:572 |

Parameters

- `callback` [function](#) A function wrapped as a selector
- `delay` [Number](#) The amount of time that the first tick will wait before execution.

Examples

```
var timeCallback = function (dt) {
    cc.log("time: " + dt);
}
this.scheduleOnce(timeCallback, 2);
```

unschedule

Unschedules a custom callback function.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:589 |

Parameters

- `callback_fn` [function](#) A function wrapped as a selector

Examples

```
this.unschedule(_callback);
```

unscheduleAllCallbacks

unschedule all scheduled callback functions: custom callback functions, and the 'update' callback function.

Actions are not affected by this method.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:605 |

Examples

```
this.unscheduleAllCallbacks();
```

destroy

Destroy this Object, and release all its own references to other objects.

Actual object destruction will delayed until before rendering. From the next frame, this object is not usable any more. You can use cc.isValid(obj) to check whether the object is destroyed before accessing it.

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:296 |

Examples

```
obj.destroy();
```

_destruct

Clear all references in the instance.

NOTE: this method will not clear the getter or setter functions which defined in the instance of CCObject. You can override the _destruct method if you need, for example:

```
_destruct: function () { for (var key in this) { if (this.hasOwnProperty(key)) { switch (typeof this[key]) { case 'string': this[key] = ''; break; case 'object': case 'function': this[key] = null; break; } } }}
```

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:430 |

_onPreDestroy

Called before the object being destroyed.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:463 |

_serialize

The customized serialization for this object. (Editor Only)

| meta | description |
|------------|---|
| Returns | object |
| Defined in | cocos2d/core/platform/CCObject.js:488 |

Parameters

- `exporting Boolean`

_deserialize

Init this object from the custom serialized data.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:498 |

Parameters

- `data Object` the serialized json data
- `ctx _Deserializer`

Canvas Class

Extends [Component](#)

Module: [cc](#)

Index

Properties

- `instance Canvas` Current active canvas, the scene should only have one active canvas at the same time.
- `designResolution Size` The design resolution for current scene.
- `fitHeight Boolean` TODO
- `fitWidth Boolean` TODO
- `__eventTargets Array` Register all related EventTargets,...
- `node Node` The node this component is attached to.
- `uuid String` The uuid for editor.
- `_enabled Boolean`
- `enabled Boolean` indicates whether this component is enabled or not.
- `enabledInHierarchy Boolean` indicates whether this component is enabled and its node is also active in the hierarchy.
- `_isOnLoadCalled Number` Returns a value which used to indicate the onLoad get called or not.
- `_name String`
- `_objFlags Number`
- `name String` The name of the object.
- `isValid Boolean` Indicates whether the object is not yet destroyed.

Methods

- `update` This is a lifecycle method.
- `lateUpdate` This is a lifecycle method.
- `__preload` __preload is called before every onLoad.
- `onLoad` When attaching to an active node or its node first activated.
- `start` Called before all scripts' update if the Component is enabled the first time.
- `onEnable` This is a lifecycle method.
- `onDisable` This is a lifecycle method.
- `onDestroy` This is a lifecycle method.
- `onFocusInEditor`
- `onLostFocusInEditor`
- `resetInEditor` Called to initialize the component or node's properties when adding the component the first time or when the Reset command is used.
- `addComponent` Adds a component class to the node.
- `getComponent` Returns the component of supplied type if the node has one attached, null if it doesn't....
- `getComponents` Returns all components of supplied Type in the node.
- `getComponentInChildren` Returns the component of supplied type in any of its children using depth first search.
- `getComponentsInChildren` Returns the components of supplied type in self or any of its children using depth first search.
- `_getLocalBounds` If the component's bounding box is different from the node's, you can implement this method to supply
- `onRestore` for undo/redo operation.
- `schedule` Schedules a custom selector....
- `scheduleOnce` Schedules a callback function that runs only once, with a delay of 0 or larger.
- `unschedule` Unschedules a custom callback function.
- `unscheduleAllCallbacks` unschedule all scheduled callback functions: custom callback functions, and the 'update' callback function....
- `destroy` Actual object destruction will delayed until before rendering.
- `_destruct` Clear all references in the instance.

- `_onPreDestroy` Called before the object being destroyed.
- `_serialize` The customized serialization for this object.
- `_deserialize` Init this object from the custom serialized data.

Details

Properties

instance

Current active canvas, the scene should only have one active canvas at the same time.

| meta | description |
|------------|--|
| Type | Canvas |
| Defined in | cocos2d/core/components/CCCanvas.js:56 |

designResolution

The design resolution for current scene.

| meta | description |
|------------|--|
| Type | Size |
| Defined in | cocos2d/core/components/CCCanvas.js:67 |

fitHeight

TODO

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCCanvas.js:89 |

fitWidth

TODO

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCCanvas.js:108 |

__eventTargets

Register all related EventTargets, all event callbacks will be removed in `_onPreDestroy`

| meta | description |
|------------|---|
| Type | Array |
| Defined in | cocos2d/core/components/CCComponent.js:61 |

node

The node this component is attached to. A component is always attached to a node.

| meta | description |
|------------|---|
| Type | Node |
| Defined in | cocos2d/core/components/CCComponent.js:75 |

Examples

```
cc.log(comp.node);
```

uuid

The uuid for editor.

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/components/CCComponent.js:106 |

Examples

```
cc.log(comp.uuid);
```

_enabled

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:147 |

enabled

indicates whether this component is enabled or not.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:154 |

Examples

```
comp.enabled = true;  
cc.log(comp.enabled);
```

enabledInHierarchy

indicates whether this component is enabled and its node is also active in the hierarchy.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:185 |

Examples

```
cc.log(comp.enabledInHierarchy);
```

_isOnLoadCalled

Returns a value which used to indicate the onLoad get called or not.

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/components/CCComponent.js:201 |

Examples

```
cc.log(this._isOnLoadCalled > 0);
```

_name

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:76 |

_objFlags

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/platform/CCObject.js:83 |

name

The name of the object.

| meta | description |
|------|------------------------|
| Type | String |

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:243 |

Examples

```
obj.name = "New Obj";
```

isValid

Indicates whether the object is not yet destroyed. (It will not be available after being destroyed)
When an object's `destroy` is called, it is actually destroyed after the end of this frame.
So `isValid` will return false from the next frame, while `isValid` in the current frame will still be true.
If you want to determine whether the current frame has called `destroy`, use `cc.isValid(obj, true)`, but this is often caused by a particular logical requirements, which is not normally required.

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:261 |

Examples

```
var node = new cc.Node();
cc.log(node.isValid);      // true
node.destroy();
cc.log(node.isValid);      // true, still valid in this frame
// after a frame...
cc.log(node.isValid);      // false, destroyed in the end of last frame
```

Methods

update

Update is called every frame, if the Component is enabled.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:222 |

Parameters

- `dt` Number the delta time in seconds it took to complete the last frame

lateUpdate

LateUpdate is called every frame, if the Component is enabled.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:233 |

__preload

`__preload` is called before every `onLoad`. It is used to initialize the builtin components internally, to avoid checking whether `onLoad` is called before every public method calls. This method should be removed if script priority is supported.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:243 |

onLoad

When attaching to an active node or its node first activated. `onLoad` is always called before any start functions, this allows you to order initialization of scripts.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:254 |

start

Called before all scripts' update if the Component is enabled the first time. Usually used to initialize some logic which need to be called after all components' onload methods called.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:267 |

onEnable

Called when this component becomes enabled and its node is active.
 This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:280 |

onDisable

Called when this component becomes disabled or its node becomes inactive.
 This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:290 |

onDestroy

Called when this component will be destroyed.
 This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:300 |

onFocusInEditor

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:310 |

onLostFocusInEditor

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:315 |

resetInEditor

Called to initialize the component or node's properties when adding the component the first time or when the Reset command is used. This function is only called in editor.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:320 |

addComponent

Adds a component class to the node. You can also add component to node by passing in the name of the script.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:330 |

Parameters

- typeOrClassName [Function](#) | [String](#) the constructor or the class name of the component to add

Examples

```
var sprite = node.addComponent(cc.Sprite);
var test = node.addComponent("Test");
```

getComponent

Returns the component of supplied type if the node has one attached, null if it doesn't. You can also get component in the node by passing in the name of the script.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:348 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
// get sprite component.
var sprite = node.getComponent(cc.Sprite);
// get custom test calss.
var test = node.getComponent("Test");
```

getComponents

Returns all components of supplied Type in the node.

| meta | description |
|------------|--|
| Returns | Component[] |
| Defined in | cocos2d/core/components/CCComponent.js:372 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
var sprites = node.getComponents(cc.Sprite);
var tests = node.getComponents("Test");
```

getComponentInChildren

Returns the component of supplied type in any of its children using depth first search.

| meta | description |
|---------|---------------------------|
| Returns | Component |

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:390 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
var sprite = node.getComponentInChildren(cc.Sprite);
var Test = node.getComponentInChildren("Test");
```

getComponentsInChildren

Returns the components of supplied type in self or any of its children using depth first search.

| meta | description |
|------------|--|
| Returns | Component[] |
| Defined in | cocos2d/core/components/CCComponent.js:408 |

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
var sprites = node.getComponentsInChildren(cc.Sprite);
var tests = node.getComponentsInChildren("Test");
```

_getLocalBounds

If the component's bounding box is different from the node's, you can implement this method to supply a custom axis aligned bounding box (AABB), so the editor's scene view can perform hit test properly.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:428 |

Parameters

- `out_rect Rect` the Rect to receive the bounding box

onRestore

onRestore is called after the user clicks the Reset item in the Inspector's context menu or performs an undo operation on this component.

If the component contains the "internal state", short for "temporary member variables which not included in its CCClass properties", then you may need to implement this function.

The editor will call the getset accessors of your component to record/restore the component's state for undo/redo operation. However, in extreme cases, it may not work well. Then you should implement this function to manually synchronize your component's "internal states" with its public properties. Once you implement this function, all the getset accessors of your component will not be called when the user performs an undo/redo operation. Which means that only the properties with default value will be recorded or restored by editor.

Similarly, the editor may fail to reset your component correctly in extreme cases. Then if you need to support the reset menu, you should manually synchronize your component's "internal states" with its properties in this function. Once you implement this function, all the getset accessors of your component will not be called during reset operation. Which means that only the properties with default value will be reset by editor.

This function is only called in editor mode.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:441 |

schedule

Schedules a custom selector.

If the selector is already scheduled, then the interval parameter will be updated without scheduling it again.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:535 |

Parameters

- `callback` [function](#) The callback function
- `interval` [Number](#) Tick interval in seconds. 0 means tick every frame.
- `repeat` [Number](#) The selector will be executed (repeat + 1) times, you can use cc.macro.REPEAT_FOREVER for tick infinitely.
- `delay` [Number](#) The amount of time that the first tick will wait before execution.

Examples

```
var timeCallback = function (dt) {
    cc.log("time: " + dt);
}
this.schedule(timeCallback, 1);
```

scheduleOnce

Schedules a callback function that runs only once, with a delay of 0 or larger.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:572 |

Parameters

- `callback` [function](#) A function wrapped as a selector
- `delay` [Number](#) The amount of time that the first tick will wait before execution.

Examples

```
var timeCallback = function (dt) {
    cc.log("time: " + dt);
}
this.scheduleOnce(timeCallback, 2);
```

unschedule

Unschedules a custom callback function.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:589 |

Parameters

- `callback_fn` [function](#) A function wrapped as a selector

Examples

```
this.unschedule(_callback);
```

unscheduleAllCallbacks

unschedule all scheduled callback functions: custom callback functions, and the 'update' callback function.

Actions are not affected by this method.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:605 |

Examples

```
this.unscheduleAllCallbacks();
```

destroy

Destroy this Object, and release all its own references to other objects.

Actual object destruction will delayed until before rendering. From the next frame, this object is not usable any more. You can use `cc.isValid(obj)` to check whether the object is destroyed before accessing it.

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:296 |

Examples

```
obj.destroy();
```

_destruct

Clear all references in the instance.

NOTE: this method will not clear the getter or setter functions which defined in the instance of CCObject. You can override the _destruct method if you need, for example: _destruct: function () { for (var key in this) { if (this.hasOwnProperty(key)) { switch (typeof this[key]) { case 'string': this[key] = ""; break; case 'object': case 'function': this[key] = null; break; } } }}

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:430 |

_onPreDestroy

Called before the object being destroyed.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:463 |

_serialize

The customized serialization for this object. (Editor Only)

| meta | description |
|------------|---|
| Returns | object |
| Defined in | cocos2d/core/platform/CCObject.js:488 |

Parameters

- exporting [Boolean](#)

_deserialize

Init this object from the custom serialized data.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:498 |

Parameters

- `data Object` the serialized json data
- `ctx _Deserializer`

CircleCollider Class

Extends [Collider](#), [Collider.Circle](#)(mixin)

Module: [cc](#)

Circle Collider.

Index

Properties

- `offset Vec2` Position offset
- `radius Number` Circle radius
- `tag Integer` Tag.
- `__eventTargets Array` Register all related EventTargets,...
- `node Node` The node this component is attached to.
- `uuid String` The uuid for editor.
- `_enabled Boolean`
- `enabled Boolean` indicates whether this component is enabled or not.
- `enabledInHierarchy Boolean` indicates whether this component is enabled and its node is also active in the hierarchy.
- `_isOnLoadCalled Number` Returns a value which used to indicate the onLoad get called or not.
- `_name String`
- `_objFlags Number`
- `name String` The name of the object.
- `isValid Boolean` Indicates whether the object is not yet destroyed.

Methods

- `update` This is a lifecycle method.
- `lateUpdate` This is a lifecycle method.
- `__preload` __preload is called before every onLoad.
- `onLoad` When attaching to an active node or its node first activated.
- `start` Called before all scripts' update if the Component is enabled the first time.
- `onEnable` This is a lifecycle method.
- `onDisable` This is a lifecycle method.

- `onDestroy` This is a lifecycle method.
- `onFocusInEditor`
- `onLostFocusInEditor`
- `resetInEditor` Called to initialize the component or node's properties when adding the component the first time or when the Reset command is used.
- `addComponent` Adds a component class to the node.
- `getComponent` Returns the component of supplied type if the node has one attached, null if it doesn't....
- `getComponents` Returns all components of supplied Type in the node.
- `getComponentInChildren` Returns the component of supplied type in any of its children using depth first search.
- `getComponentsInChildren` Returns the components of supplied type in self or any of its children using depth first search.
- `_getLocalBounds` If the component's bounding box is different from the node's, you can implement this method to supply
- `onRestore` for undo/redo operation.
- `schedule` Schedules a custom selector....
- `scheduleOnce` Schedules a callback function that runs only once, with a delay of 0 or larger.
- `unschedule` Unschedules a custom callback function.
- `unscheduleAllCallbacks` unschedule all scheduled callback functions: custom callback functions, and the 'update' callback function....
- `destroy` Actual object destruction will delayed until before rendering.
- `_destruct` Clear all references in the instance.
- `_onPreDestroy` Called before the object being destroyed.
- `_serialize` The customized serialization for this object.
- `_deserialize` Init this object from the custom serialized data.

Details

Properties

`offset`

Position offset

| meta | description |
|------------|--|
| Type | Vec2 |
| Defined in | cocos2d/core/collider/CCCircleCollider.js:40 |

`radius`

Circle radius

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/collider/CCCircleCollider.js:56 |

tag

Tag. If a node has several collider components, you can judge which type of collider is collided according to the tag.

| meta | description |
|------------|--|
| Type | Integer |
| Defined in | cocos2d/core/collider/CCCollider.js:47 |

__eventTargets

Register all related EventTargets, all event callbacks will be removed in _onPreDestroy

| meta | description |
|------------|---|
| Type | Array |
| Defined in | cocos2d/core/components/CCComponent.js:61 |

node

The node this component is attached to. A component is always attached to a node.

| meta | description |
|------------|---|
| Type | Node |
| Defined in | cocos2d/core/components/CCComponent.js:75 |

Examples

```
cc.log(comp.node);
```

uuid

The uuid for editor.

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/components/CCComponent.js:106 |

Examples

```
cc.log(comp.uuid);
```

_enabled

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:147 |

enabled

indicates whether this component is enabled or not.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:154 |

Examples

```
comp.enabled = true;  
cc.log(comp.enabled);
```

enabledInHierarchy

indicates whether this component is enabled and its node is also active in the hierarchy.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:185 |

Examples

```
cc.log(comp.enabledInHierarchy);
```

_isOnLoadCalled

Returns a value which used to indicate the onLoad get called or not.

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/components/CCComponent.js:201 |

Examples

```
cc.log(this._isOnLoadCalled > 0);
```

_name

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:76 |

_objFlags

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/platform/CCObject.js:83 |

name

The name of the object.

| meta | description |
|------------|---|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:243 |

Examples

```
obj.name = "New Obj";
```

isValid

Indicates whether the object is not yet destroyed. (It will not be available after being destroyed)
When an object's `destroy` is called, it is actually destroyed after the end of this frame.
So `isValid` will return false from the next frame, while `isValid` in the current frame will still be true.
If you want to determine whether the current frame has called `destroy`, use `cc.isValid(obj, true)`, but this is often caused by a particular logical requirements, which is not normally required.

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:261 |

Examples

```
var node = new cc.Node();
cc.log(node.isValid);      // true
node.destroy();
cc.log(node.isValid);      // true, still valid in this frame
// after a frame...
cc.log(node.isValid);      // false, destroyed in the end of last frame
```

Methods

update

Update is called every frame, if the Component is enabled.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:222 |

Parameters

- `dt` Number the delta time in seconds it took to complete the last frame

lateUpdate

LateUpdate is called every frame, if the Component is enabled.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:233 |

__preload

__preload is called before every onLoad. It is used to initialize the builtin components internally, to avoid checking whether onLoad is called before every public method calls. This method should be removed if script priority is supported.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:243 |

onLoad

When attaching to an active node or its node first activated. onLoad is always called before any start functions, this allows you to order initialization of scripts.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:254 |

start

Called before all scripts' update if the Component is enabled the first time. Usually used to initialize some logic which need to be called after all components' `onload` methods called. This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:267 |

onEnable

Called when this component becomes enabled and its node is active. This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:280 |

onDisable

Called when this component becomes disabled or its node becomes inactive. This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:290 |

onDestroy

Called when this component will be destroyed. This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|-------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:300 |

onFocusInEditor

| meta | description |
|-------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:310 |

onLostFocusInEditor

| meta | description |
|-------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:315 |

resetInEditor

Called to initialize the component or node's properties when adding the component the first time or when the Reset command is used. This function is only called in editor.

| meta | description |
|-------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:320 |

addComponent

Adds a component class to the node. You can also add component to node by passing in the name of the script.

| meta | description |
|-------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:330 |

Parameters

- `typeOrClassName` [Function](#) | [String](#) the constructor or the class name of the component to add

Examples

```
var sprite = node.addComponent(cc.Sprite);
var test = node.addComponent("Test");
```

getComponent

Returns the component of supplied type if the node has one attached, null if it doesn't. You can also get component in the node by passing in the name of the script.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:348 |

Parameters

- `typeOrClassName` [Function](#) | [String](#)

Examples

```
// get sprite component.
var sprite = node.getComponent(cc.Sprite);
// get custom test calss.
var test = node.getComponent("Test");
```

getComponents

Returns all components of supplied Type in the node.

| meta | description |
|------------|--|
| Returns | Component[] |
| Defined in | cocos2d/core/components/CCComponent.js:372 |

Parameters

- `typeOrClassName` [Function](#) | [String](#)

Examples

```
var sprites = node.getComponents(cc.Sprite);
var tests = node.getComponents("Test");
```

getComponentInChildren

Returns the component of supplied type in any of its children using depth first search.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:390 |

Parameters

- `typeOrClassName` [Function](#) | [String](#)

Examples

```
var sprite = node.getComponentInChildren(cc.Sprite);
var Test = node.getComponentInChildren("Test");
```

getComponentsInChildren

Returns the components of supplied type in self or any of its children using depth first search.

| meta | description |
|------------|--|
| Returns | Component[] |
| Defined in | cocos2d/core/components/CCComponent.js:408 |

Parameters

- `typeOrClassName` [Function](#) | [String](#)

Examples

```
var sprites = node.getComponentsInChildren(cc.Sprite);
var tests = node.getComponentsInChildren("Test");
```

`_getLocalBounds`

If the component's bounding box is different from the node's, you can implement this method to supply a custom axis aligned bounding box (AABB), so the editor's scene view can perform hit test properly.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:428 |

Parameters

- `out_rect Rect` the Rect to receive the bounding box

onRestore

`onRestore` is called after the user clicks the Reset item in the Inspector's context menu or performs an undo operation on this component.

If the component contains the "internal state", short for "temporary member variables which not included in its CCClass properties", then you may need to implement this function.

The editor will call the getset accessors of your component to record/restore the component's state for undo/redo operation. However, in extreme cases, it may not work well. Then you should implement this function to manually synchronize your component's "internal states" with its public properties. Once you implement this function, all the getset accessors of your component will not be called when the user performs an undo/redo operation. Which means that only the properties with default value will be recorded or restored by editor.

Similarly, the editor may fail to reset your component correctly in extreme cases. Then if you need to support the reset menu, you should manually synchronize your component's "internal states" with its properties in this function. Once you implement this function, all the getset accessors of your component will not be called during reset operation. Which means that only the properties with default value will be reset by editor.

This function is only called in editor mode.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:441 |

schedule

Schedules a custom selector.

If the selector is already scheduled, then the interval parameter will be updated without scheduling it again.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:535 |

Parameters

- `callback` [function](#) The callback function
- `interval` [Number](#) Tick interval in seconds. 0 means tick every frame.
- `repeat` [Number](#) The selector will be executed (repeat + 1) times, you can use cc.macro.REPEAT_FOREVER for tick infinitely.
- `delay` [Number](#) The amount of time that the first tick will wait before execution.

Examples

```
var timeCallback = function (dt) {
    cc.log("time: " + dt);
}
this.schedule(timeCallback, 1);
```

scheduleOnce

Schedules a callback function that runs only once, with a delay of 0 or larger.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:572 |

Parameters

- `callback` [function](#) A function wrapped as a selector
- `delay` [Number](#) The amount of time that the first tick will wait before execution.

Examples

```
var timeCallback = function (dt) {  
    cc.log("time: " + dt);  
}  
this.scheduleOnce(timeCallback, 2);
```

unschedule

Unschedules a custom callback function.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:589 |

Parameters

- `callback_fn` [function](#) A function wrapped as a selector

Examples

```
this.unschedule(_callback);
```

unscheduleAllCallbacks

unschedule all scheduled callback functions: custom callback functions, and the 'update' callback function.

Actions are not affected by this method.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:605 |

Examples

```
this.unscheduleAllCallbacks();
```

destroy

Destroy this Object, and release all its own references to other objects.

Actual object destruction will delayed until before rendering. From the next frame, this object is not usable any more. You can use `cc.isValid(obj)` to check whether the object is destroyed before accessing it.

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:296 |

Examples

```
obj.destroy();
```

_destruct

Clear all references in the instance.

NOTE: this method will not clear the getter or setter functions which defined in the instance of CCObject. You can override the _destruct method if you need, for example: _destruct: function () { for (var key in this) { if (this.hasOwnProperty(key)) { switch (typeof this[key]) { case 'string': this[key] = ''; break; case 'object': case 'function': this[key] = null; break; } } }

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:430 |

_onPreDestroy

Called before the object being destroyed.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:463 |

_serialize

The customized serialization for this object. (Editor Only)

| meta | description |
|------------|---|
| Returns | object |
| Defined in | cocos2d/core/platform/CCObject.js:488 |

Parameters

- `exporting Boolean`

`_deserialize`

Init this object from the custom serialized data.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:498 |

Parameters

- `data Object` the serialized json data
- `ctx _Deserializer`

CoffeeScript Class

Extends [Asset](#)

Module: [cc](#)

Class for coffeescript handling.

Index

Properties

- `loaded Boolean` Whether the asset is loaded or not
- `nativeUrl String` Returns the url of this asset's native object, if none it will returns an empty string.
- `_native String` Serializable url for native asset.
- `_nativeAsset Object` The underlying native asset of this asset if one is available.
- `_uuid String`
- `_name String`
- `_objFlags Number`
- `name String` The name of the object.
- `isValid Boolean` Indicates whether the object is not yet destroyed.

Methods

- `toString` Returns the asset's url.
- `serialize` 应 AssetDB 要求提供这个方法
- `createNode` Create a new node using this asset in the scene....

- `_setRawAsset` Set native file name for this asset.
- `destroy` Actual object destruction will delayed until before rendering.
- `_destruct` Clear all references in the instance.
- `_onPreDestroy` Called before the object being destroyed.
- `_serialize` The customized serialization for this object.
- `_deserialize` Init this object from the custom serialized data.

Details

Properties

loaded

Whether the asset is loaded or not

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/assets/CCAsset.js:57 |

nativeUrl

Returns the url of this asset's native object, if none it will returns an empty string.

| meta | description |
|------------|---|
| Type | String |
| Defined in | cocos2d/core/assets/CCAsset.js:70 |

_native

Serializable url for native asset.

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/assets/CCAsset.js:108 |

_nativeAsset

The underlying native asset of this asset if one is available. This property can be used to access additional details or functionality related to the asset. This property will be initialized by the loader if `_native` is available.

| meta | description |
|-------------|--|
| Type | Object |
| Defined in | cocos2d/core/assets/CCAsset.js:116 |

_uuid

| meta | description |
|-------------|--|
| Type | String |
| Defined in | cocos2d/core/assets/CCRawAsset.js:46 |

_name

| meta | description |
|-------------|--|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:76 |

_objFlags

| meta | description |
|-------------|--|
| Type | Number |
| Defined in | cocos2d/core/platform/CCObject.js:83 |

name

The name of the object.

| meta | description |
|------------|---|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:243 |

Examples

```
obj.name = "New Obj";
```

isValid

Indicates whether the object is not yet destroyed. (It will not be available after being destroyed)
When an object's `destroy` is called, it is actually destroyed after the end of this frame.
So `isValid` will return false from the next frame, while `isValid` in the current frame will still be true.
If you want to determine whether the current frame has called `destroy`, use `cc.isValid(obj, true)`, but this is often caused by a particular logical requirements, which is not normally required.

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:261 |

Examples

```
var node = new cc.Node();
cc.log(node.isValid);      // true
node.destroy();
cc.log(node.isValid);      // true, still valid in this frame
// after a frame...
cc.log(node.isValid);      // false, destroyed in the end of last frame
```

Methods

toString

Returns the asset's url.

The Asset object overrides the `toString()` method of the Object object. For Asset objects, the `toString()` method returns a string representation of the object. JavaScript calls the `toString()` method automatically when an asset is to be represented as a text value or when a texture is referred to in a string concatenation.

| meta | description |
|------------|--|
| Returns | String |
| Defined in | cocos2d/core/assets/CCAsset.js:165 |

serialize

应 AssetDB 要求提供这个方法

| meta | description |
|------------|--|
| Returns | String |
| Defined in | cocos2d/core/assets/CCAsset.js:179 |

createNode

Create a new node using this asset in the scene.

If this type of asset dont have its corresponding node type, this method should be null.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/assets/CCAsset.js:190 |

Parameters

- callback [Function](#)
- error [String](#) null or the error info
- node [Object](#) the created node or null

_setRawAsset

Set native file name for this asset.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/assets/CCAsset.js:205 |

Parameters

- `filename` [String](#)
- `inLibrary` [Boolean](#)

destroy

Destroy this Object, and release all its own references to other objects.

Actual object destruction will delayed until before rendering. From the next frame, this object is not usable any more. You can use `cc.isValid(obj)` to check whether the object is destroyed before accessing it.

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:296 |

Examples

```
obj.destroy();
```

```
_destruct
```

Clear all references in the instance.

NOTE: this method will not clear the getter or setter functions which defined in the instance of CCObject. You can override the `_destruct` method if you need, for example:

```
_destruct: function () { for (var key in this) { if (this.hasOwnProperty(key)) { switch (typeof this[key]) { case 'string': this[key] = ""; break; case 'object': case 'function': this[key] = null; break; } } }}
```

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:430 |

```
_onPreDestroy
```

Called before the object being destroyed.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:463 |

_serialize

The customized serialization for this object. (Editor Only)

| meta | description |
|------------|---|
| Returns | object |
| Defined in | cocos2d/core/platform/CCObject.js:488 |

Parameters

- `exporting Boolean`

_deserialize

Init this object from the custom serialized data.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:498 |

Parameters

- `data Object` the serialized json data
- `ctx _Deserializer`

Collider Class

Extends [Component](#)

Module: [cc](#)

Collider component base class.

Index

Properties

- `tag Integer` Tag.
- `__eventTargets Array` Register all related EventTargets,...
- `node Node` The node this component is attached to.
- `uuid String` The uid for editor.

- `_enabled Boolean`
- `enabled Boolean` indicates whether this component is enabled or not.
- `enabledInHierarchy Boolean` indicates whether this component is enabled and its node is also active in the hierarchy.
- `_isOnLoadCalled Number` Returns a value which used to indicate the onLoad get called or not.
- `_name String`
- `_objFlags Number`
- `name String` The name of the object.
- `isValid Boolean` Indicates whether the object is not yet destroyed.

Methods

- `update` This is a lifecycle method.
- `lateUpdate` This is a lifecycle method.
- `__preload` __preload is called before every onLoad.
- `onLoad` When attaching to an active node or its node first activated.
- `start` Called before all scripts' update if the Component is enabled the first time.
- `onEnable` This is a lifecycle method.
- `onDisable` This is a lifecycle method.
- `onDestroy` This is a lifecycle method.
- `onFocusInEditor`
- `onLostFocusInEditor`
- `resetInEditor` Called to initialize the component or node's properties when adding the component the first time or when the Reset command is used.
- `addComponent` Adds a component class to the node.
- `getComponent` Returns the component of supplied type if the node has one attached, null if it doesn't....
- `getComponents` Returns all components of supplied Type in the node.
- `getComponentInChildren` Returns the component of supplied type in any of its children using depth first search.
- `getComponentsInChildren` Returns the components of supplied type in self or any of its children using depth first search.
- `_getLocalBounds` If the component's bounding box is different from the node's, you can implement this method to supply
- `onRestore` for undo/redo operation.
- `schedule` Schedules a custom selector....
- `scheduleOnce` Schedules a callback function that runs only once, with a delay of 0 or larger.
- `unschedule` Unschedules a custom callback function.
- `unscheduleAllCallbacks` unschedule all scheduled callback functions: custom callback functions, and the 'update' callback function....
- `destroy` Actual object destruction will delayed until before rendering.
- `_destruct` Clear all references in the instance.
- `_onPreDestroy` Called before the object being destroyed.
- `_serialize` The customized serialization for this object.
- `_deserialize` Init this object from the custom serialized data.

Details

Properties

tag

Tag. If a node has several collider components, you can judge which type of collider is collided according to the tag.

| meta | description |
|------------|--|
| Type | Integer |
| Defined in | cocos2d/core/collider/CCCollider.js:47 |

__eventTargets

Register all related EventTargets, all event callbacks will be removed in _onPreDestroy

| meta | description |
|------------|---|
| Type | Array |
| Defined in | cocos2d/core/components/CCComponent.js:61 |

node

The node this component is attached to. A component is always attached to a node.

| meta | description |
|------------|---|
| Type | Node |
| Defined in | cocos2d/core/components/CCComponent.js:75 |

Examples

```
cc.log(comp.node);
```

uuid

The uuid for editor.

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/components/CCComponent.js:106 |

Examples

```
cc.log(comp.uuid);
```

_enabled

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:147 |

enabled

indicates whether this component is enabled or not.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:154 |

Examples

```
comp.enabled = true;
cc.log(comp.enabled);
```

enabledInHierarchy

indicates whether this component is enabled and its node is also active in the hierarchy.

| meta | description |
|------------|--|
| Type | Boolean |
| Defined in | cocos2d/core/components/CCComponent.js:185 |

Examples

```
cc.log(comp.enabledInHierarchy);
```

_isOnLoadCalled

Returns a value which used to indicate the onLoad get called or not.

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/components/CCComponent.js:201 |

Examples

```
cc.log(this._isOnLoadCalled > 0);
```

_name

| meta | description |
|------------|--|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:76 |

_objFlags

| meta | description |
|------|------------------------|
| Type | Number |

| meta | description |
|------------|--|
| Defined in | cocos2d/core/platform/CCObject.js:83 |

name

The name of the object.

| meta | description |
|------------|---|
| Type | String |
| Defined in | cocos2d/core/platform/CCObject.js:243 |

Examples

```
obj.name = "New Obj";
```

isValid

Indicates whether the object is not yet destroyed. (It will not be available after being destroyed)
When an object's `destroy` is called, it is actually destroyed after the end of this frame.
So `isValid` will return false from the next frame, while `isValid` in the current frame will still be true.
If you want to determine whether the current frame has called `destroy`, use `cc.isValid(obj, true)`, but this is often caused by a particular logical requirements, which is not normally required.

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:261 |

Examples

```
var node = new cc.Node();
cc.log(node.isValid);    // true
node.destroy();
cc.log(node.isValid);    // true, still valid in this frame
// after a frame...
cc.log(node.isValid);    // false, destroyed in the end of last frame
```

Methods

update

Update is called every frame, if the Component is enabled.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:222 |

Parameters

- `dt` `Number` the delta time in seconds it took to complete the last frame

lateUpdate

LateUpdate is called every frame, if the Component is enabled.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:233 |

__preload

__preload is called before every onLoad. It is used to initialize the builtin components internally, to avoid checking whether onLoad is called before every public method calls. This method should be removed if script priority is supported.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:243 |

onLoad

When attaching to an active node or its node first activated. onLoad is always called before any start functions, this allows you to order initialization of scripts.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:254 |

start

Called before all scripts' update if the Component is enabled the first time. Usually used to initialize some logic which need to be called after all components' `onload` methods called. This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:267 |

onEnable

Called when this component becomes enabled and its node is active. This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:280 |

onDisable

Called when this component becomes disabled or its node becomes inactive. This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:290 |

onDestroy

Called when this component will be destroyed. This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

| meta | description |
|-------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:300 |

onFocusInEditor

| meta | description |
|-------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:310 |

onLostFocusInEditor

| meta | description |
|-------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:315 |

resetInEditor

Called to initialize the component or node's properties when adding the component the first time or when the Reset command is used. This function is only called in editor.

| meta | description |
|-------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:320 |

addComponent

Adds a component class to the node. You can also add component to node by passing in the name of the script.

| meta | description |
|-------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:330 |

Parameters

- `typeOrClassName` [Function](#) | [String](#) the constructor or the class name of the component to add

Examples

```
var sprite = node.addComponent(cc.Sprite);
var test = node.addComponent("Test");
```

getComponent

Returns the component of supplied type if the node has one attached, null if it doesn't. You can also get component in the node by passing in the name of the script.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:348 |

Parameters

- `typeOrClassName` [Function](#) | [String](#)

Examples

```
// get sprite component.
var sprite = node.getComponent(cc.Sprite);
// get custom test calss.
var test = node.getComponent("Test");
```

getComponents

Returns all components of supplied Type in the node.

| meta | description |
|------------|--|
| Returns | Component[] |
| Defined in | cocos2d/core/components/CCComponent.js:372 |

Parameters

- `typeOrClassName` [Function](#) | [String](#)

Examples

```
var sprites = node.getComponents(cc.Sprite);
var tests = node.getComponents("Test");
```

getComponentInChildren

Returns the component of supplied type in any of its children using depth first search.

| meta | description |
|------------|--|
| Returns | Component |
| Defined in | cocos2d/core/components/CCComponent.js:390 |

Parameters

- `typeOrClassName` [Function](#) | [String](#)

Examples

```
var sprite = node.getComponentInChildren(cc.Sprite);
var Test = node.getComponentInChildren("Test");
```

getComponentsInChildren

Returns the components of supplied type in self or any of its children using depth first search.

| meta | description |
|------------|--|
| Returns | Component[] |
| Defined in | cocos2d/core/components/CCComponent.js:408 |

Parameters

- `typeOrClassName` [Function](#) | [String](#)

Examples

```
var sprites = node.getComponentsInChildren(cc.Sprite);
var tests = node.getComponentsInChildren("Test");
```

`_getLocalBounds`

If the component's bounding box is different from the node's, you can implement this method to supply a custom axis aligned bounding box (AABB), so the editor's scene view can perform hit test properly.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:428 |

Parameters

- `out_rect Rect` the Rect to receive the bounding box

onRestore

`onRestore` is called after the user clicks the Reset item in the Inspector's context menu or performs an undo operation on this component.

If the component contains the "internal state", short for "temporary member variables which not included in its CCClass properties", then you may need to implement this function.

The editor will call the getset accessors of your component to record/restore the component's state for undo/redo operation. However, in extreme cases, it may not work well. Then you should implement this function to manually synchronize your component's "internal states" with its public properties. Once you implement this function, all the getset accessors of your component will not be called when the user performs an undo/redo operation. Which means that only the properties with default value will be recorded or restored by editor.

Similarly, the editor may fail to reset your component correctly in extreme cases. Then if you need to support the reset menu, you should manually synchronize your component's "internal states" with its properties in this function. Once you implement this function, all the getset accessors of your component will not be called during reset operation. Which means that only the properties with default value will be reset by editor.

This function is only called in editor mode.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:441 |

schedule

Schedules a custom selector.

If the selector is already scheduled, then the interval parameter will be updated without scheduling it again.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:535 |

Parameters

- `callback function` The callback function
- `interval Number` Tick interval in seconds. 0 means tick every frame.
- `repeat Number` The selector will be executed (repeat + 1) times, you can use cc.macro.REPEAT_FOREVER for tick infinitely.
- `delay Number` The amount of time that the first tick will wait before execution.

Examples

```
var timeCallback = function (dt) {
    cc.log("time: " + dt);
}
this.schedule(timeCallback, 1);
```

scheduleOnce

Schedules a callback function that runs only once, with a delay of 0 or larger.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:572 |

Parameters

- `callback function` A function wrapped as a selector
- `delay Number` The amount of time that the first tick will wait before execution.

Examples

```
var timeCallback = function (dt) {  
    cc.log("time: " + dt);  
}  
this.scheduleOnce(timeCallback, 2);
```

unschedule

Unschedules a custom callback function.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:589 |

Parameters

- `callback_fn` [function](#) A function wrapped as a selector

Examples

```
this.unschedule(_callback);
```

unscheduleAllCallbacks

unschedule all scheduled callback functions: custom callback functions, and the 'update' callback function.

Actions are not affected by this method.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/components/CCComponent.js:605 |

Examples

```
this.unscheduleAllCallbacks();
```

destroy

Destroy this Object, and release all its own references to other objects.

Actual object destruction will delayed until before rendering. From the next frame, this object is not usable any more. You can use `cc.isValid(obj)` to check whether the object is destroyed before accessing it.

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/platform/CCObject.js:296 |

Examples

```
obj.destroy();
```

_destruct

Clear all references in the instance.

NOTE: this method will not clear the getter or setter functions which defined in the instance of CCObject. You can override the _destruct method if you need, for example: _destruct: function () { for (var key in this) { if (this.hasOwnProperty(key)) { switch (typeof this[key]) { case 'string': this[key] = ''; break; case 'object': case 'function': this[key] = null; break; } } }

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:430 |

_onPreDestroy

Called before the object being destroyed.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:463 |

_serialize

The customized serialization for this object. (Editor Only)

| meta | description |
|------------|---|
| Returns | object |
| Defined in | cocos2d/core/platform/CCObject.js:488 |

Parameters

- `exporting Boolean`

_deserialize

Init this object from the custom serialized data.

| meta | description |
|------------|---|
| Defined in | cocos2d/core/platform/CCObject.js:498 |

Parameters

- `data Object` the serialized json data
- `ctx _Deserializer`

Collider.Box Class

Module: [cc](#)

Defines a Box Collider .

Index

Properties

- `offset Vec2` Position offset
- `size Size` Box size

Details

Properties

offset

Position offset

| meta | description |
|------|----------------------|
| Type | Vec2 |

| meta | description |
|------------|---|
| Defined in | cocos2d/core/collider/CCBoxCollider.js:41 |

size

Box size

| meta | description |
|------------|---|
| Type | Size |
| Defined in | cocos2d/core/collider/CCBoxCollider.js:58 |

Collider.Circle Class

Module: [cc](#)

Defines a Circle Collider .

Index

Properties

- `offset` Vec2 Position offset
- `radius` Number Circle radius

Details

Properties

offset

Position offset

| meta | description |
|------|----------------------|
| Type | Vec2 |

| meta | description |
|------------|--|
| Defined in | cocos2d/core/collider/CCCircleCollider.js:40 |

radius

Circle radius

| meta | description |
|------------|--|
| Type | Number |
| Defined in | cocos2d/core/collider/CCCircleCollider.js:56 |

Collider.Polygon Class

Defined in: <https://github.com/cocos-creator/engine/blob/4f734a806d1fd7c4073fb064fddc961384fe67af/cocos2d/core/collider/CCPolygonCollider.js:30>

Module: [cc](#)

Defines a Polygon Collider .

Index

Properties

- [offset](#) Vec2 Position offset
- [points](#) Vec2[] Polygon points

Details

Properties

[offset](#)

Position offset

| meta | description |
|------------|---|
| Type | Vec2 |
| Defined in | cocos2d/core/collider/CCPolygonCollider.js:45 |

points

Polygon points

| meta | description |
|------------|---|
| Type | Vec2[] |
| Defined in | cocos2d/core/collider/CCPolygonCollider.js:61 |

CollisionManager Class

Module: [cc](#)

A simple collision manager class. It will calculate whether the collider collides other colliders, if collides then call the callbacks.

Index

Properties

- `enabled Boolean`
- `enabledDrawBoundingBox Boolean`
- `enabledDebugDraw Boolean`

Methods

- `hasEventListener` Checks whether the EventTarget object has any callback registered for a specific type of event.
- `on` Register an callback of a specific event type on the EventTarget.
- `off` Removes the listeners previously registered with the same type, callback, target and or useCapture,...
- `targetOff` Removes all callbacks previously registered with the same target (passed as parameter).
- `once` Register an callback of a specific event type on the EventTarget,...
- `emit` Trigger an event directly with the event name and necessary arguments.

- [dispatchEvent](#) Send an event with the event object.

Details

Properties

enabled

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/collider/CCCollisionManager.js:120 |

enabledDrawBoundingBox

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/collider/CCCollisionManager.js:128 |

enabledDebugDraw

| meta | description |
|------------|---|
| Type | Boolean |
| Defined in | cocos2d/core/collider/CCCollisionManager.js:462 |

Methods

hasEventListener

Checks whether the EventTarget object has any callback registered for a specific type of event.

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/event/event-target.js:68 |

Parameters

- type [String](#) The type of event.

on

Register an callback of a specific event type on the EventTarget. This type of event should be triggered via `emit`.

| meta | description |
|------------|---|
| Returns | Function |
| Defined in | cocos2d/core/event/event-target.js:76 |

Parameters

- type [String](#) A string representing the event type to listen for.
- callback [Function](#) The callback that will be invoked when the event is dispatched.
- The callback is ignored if it is a duplicate (the callbacks are unique).
- arg1 Any arg1
- arg2 Any arg2
- arg3 Any arg3
- arg4 Any arg4
- arg5 Any arg5
- target [Object](#) The target (this object) to invoke the callback, can be null

Examples

```
eventTarget.on('fire', function (event) {
    cc.log("fire in the hole");
}, node);
```

off

Removes the listeners previously registered with the same type, callback, target and or `useCapture`, if only type is passed as parameter, all listeners registered with that type will be removed.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/event/event-target.js:117 |

Parameters

- type [String](#) A string representing the event type being removed.
- callback [Function](#) The callback to remove.
- target [Object](#) The target (this object) to invoke the callback, if it's not given, only callback without target will be removed

Examples

```
// register fire eventListener
var callback = eventTarget.on('fire', function (event) {
    cc.log("fire in the hole");
}, target);
// remove fire event listener
eventTarget.off('fire', callback, target);
// remove all fire event listeners
eventTarget.off('fire');
```

targetOff

Removes all callbacks previously registered with the same target (passed as parameter). This is not for removing all listeners in the current event target, and this is not for removing all listeners the target parameter have registered. It's only for removing all listeners (callback and target couple) registered on the current event target by the target parameter.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/event/event-target.js:151 |

Parameters

- target [Object](#) The target to be searched for all related listeners

once

Register an callback of a specific event type on the EventTarget, the callback will remove itself after the first time it is triggered.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/event/event-target.js:164 |

Parameters

- type [String](#) A string representing the event type to listen for.
- callback [Function](#) The callback that will be invoked when the event is dispatched.
 - The callback is ignored if it is a duplicate (the callbacks are unique).
- arg1 Any arg1
- arg2 Any arg2
- arg3 Any arg3
- arg4 Any arg4
- arg5 Any arg5
- target [Object](#) The target (this object) to invoke the callback, can be null

Examples

```
eventTarget.once('fire', function (event) {
    cc.log("this is the callback and will be invoked only once");
}, node);
```

emit

Trigger an event directly with the event name and necessary arguments.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/event/event-target.js:201 |

Parameters

- type [String](#) event type
- arg1 Any First argument
- arg2 Any Second argument
- arg3 Any Third argument
- arg4 Any Fourth argument
- arg5 Any Fifth argument

Examples

```
eventTarget.emit('fire', event);
eventTarget.emit('fire', message, emitter);
```

dispatchEvent

Send an event with the event object.

| meta | description |
|------------|--|
| Defined in | cocos2d/core/event/event-target.js:221 |

Parameters

- event [Event](#)

Color Class

Extends [ValueType](#)

Module: [cc](#)

Representation of RGBA colors.

Each color component is a floating point value with a range from 0 to 255.

You can also use the convenience method [cc.color](#) to create a new Color.

Index

Properties

- [WHITE](#) Color Solid white, RGBA is [255, 255, 255, 255].
- [BLACK](#) Color Solid black, RGBA is [0, 0, 0, 255].
- [TRANSPARENT](#) Color Transparent, RGBA is [0, 0, 0, 0].
- [GRAY](#) Color Grey, RGBA is [127.5, 127.5, 127.5].
- [RED](#) Color Solid red, RGBA is [255, 0, 0].
- [GREEN](#) Color Solid green, RGBA is [0, 255, 0].
- [BLUE](#) Color Solid blue, RGBA is [0, 0, 255].
- [YELLOW](#) Color Yellow, RGBA is [255, 235, 4].
- [ORANGE](#) Color Orange, RGBA is [255, 127, 0].
- [CYAN](#) Color Cyan, RGBA is [0, 255, 255].
- [MAGENTA](#) Color Magenta, RGBA is [255, 0, 255].

Methods

- [constructor](#)
- [clone](#) Clone a new color from the current color.
- [equals](#) TODO
- [lerp](#) TODO
- [toString](#) TODO
- [getR](#) Gets red channel value
- [setR](#) Sets red value and return the current color object
- [getG](#) Gets green channel value
- [setG](#) Sets green value and return the current color object
- [getB](#) Gets blue channel value
- [setB](#) Sets blue value and return the current color object
- [getA](#) Gets alpha channel value
- [setA](#) Sets alpha value and return the current color object
- [toCSS](#) Convert color to css format.

- [fromHEX](#) Read hex string and store color data into the current color object, the hex string must be formated as rgba or rgb.
- [toHEX](#) convert Color to HEX color string.
- [toRGBValue](#) Convert to 24bit rgb value.
- [fromHSV](#) Read HSV model color and convert to RGB color
- [toHSV](#) Transform to HSV model color
- [set](#) Copies all the properties from another given object to this value.

Details

Properties

WHITE

Solid white, RGBA is [255, 255, 255, 255].

| meta | description |
|------------|--|
| Type | Color |
| Defined in | cocos2d/core/value-types/color.js:78 |

BLACK

Solid black, RGBA is [0, 0, 0, 255].

| meta | description |
|------------|--|
| Type | Color |
| Defined in | cocos2d/core/value-types/color.js:86 |

TRANSPARENT

Transparent, RGBA is [0, 0, 0, 0].

| meta | description |
|------------|--|
| Type | Color |
| Defined in | cocos2d/core/value-types/color.js:94 |

GRAY

Grey, RGBA is [127.5, 127.5, 127.5].

| meta | description |
|------------|---|
| Type | Color |
| Defined in | cocos2d/core/value-types/color.js:102 |

RED

Solid red, RGBA is [255, 0, 0].

| meta | description |
|------------|---|
| Type | Color |
| Defined in | cocos2d/core/value-types/color.js:110 |

GREEN

Solid green, RGBA is [0, 255, 0].

| meta | description |
|------------|---|
| Type | Color |
| Defined in | cocos2d/core/value-types/color.js:118 |

BLUE

Solid blue, RGBA is [0, 0, 255].

| meta | description |
|------------|---|
| Type | Color |
| Defined in | cocos2d/core/value-types/color.js:126 |

YELLOW

Yellow, RGBA is [255, 235, 4].

| meta | description |
|------------|---|
| Type | Color |
| Defined in | cocos2d/core/value-types/color.js:134 |

ORANGE

Orange, RGBA is [255, 127, 0].

| meta | description |
|------------|---|
| Type | Color |
| Defined in | cocos2d/core/value-types/color.js:142 |

CYAN

Cyan, RGBA is [0, 255, 255].

| meta | description |
|------------|---|
| Type | Color |
| Defined in | cocos2d/core/value-types/color.js:150 |

MAGENTA

Magenta, RGBA is [255, 0, 255].

| meta | description |
|------------|---|
| Type | Color |
| Defined in | cocos2d/core/value-types/color.js:158 |

Methods

constructor

| meta | description |
|------------|--|
| Defined in | cocos2d/core/value-types/color.js:53 |

Parameters

- r [Number](#) red component of the color, default value is 0.
- g [Number](#) green component of the color, defualt value is 0.
- b [Number](#) blue component of the color, default value is 0.
- a [Number](#) alpha component of the color, default value is 255.

clone

Clone a new color from the current color.

| meta | description |
|------------|---|
| Returns | Color |
| Defined in | cocos2d/core/value-types/color.js:177 |

Examples

```
var color = new cc.Color();
var newColor = color.clone(); // Color {r: 0, g: 0, b: 0, a: 255}
```

equals

TODO

| meta | description |
|------------|---|
| Returns | Boolean |
| Defined in | cocos2d/core/value-types/color.js:192 |

Parameters

- other [Color](#)

Examples

```
var color1 = cc.Color.WHITE;
var color2 = new cc.Color(255, 255, 255);
cc.log(color1.equals(color2)); // true;
color2 = cc.Color.RED;
cc.log(color2.equals(color1)); // false;
```

lerp

TODO

| meta | description |
|------------|---|
| Returns | Color |
| Defined in | cocos2d/core/value-types/color.js:209 |

Parameters

- [to Color](#)
- ratio [number](#) the interpolation coefficient.
- out [Color](#) optional, the receiving vector.

Examples

```
```Not found for the example path: temp-
src/engine/docs/api/engine/docs/cocos2d/core/value-types/CCColor/lerp.js
```

## toString

### TODO

meta	description
Returns	<a href="#">String</a>
Defined in	<a href="#">cocos2d/core/value-types/color.js:232</a>

## Examples

```
var color = cc.Color.WHITE;
```

```
color.toString(); // "rgba(255, 255, 255, 255)"
```

## getR

Gets red channel value

meta	description
Returns	<a href="#">Number</a>
Defined in	<a href="#">cocos2d/core/value-types/color.js:249</a>

## setR

Sets red value and return the current color object

meta	description
Returns	<a href="#">Color</a>
Defined in	<a href="#">cocos2d/core/value-types/color.js:258</a>

Parameters

- `red` [Number](#) the new Red component.

Examples

```
var color = new cc.Color();
color.setR(255); // Color {r: 255, g: 0, b: 0, a: 255}
```

## getG

Gets green channel value

meta	description
Returns	<a href="#">Number</a>
Defined in	<a href="#">cocos2d/core/value-types/color.js:273</a>

## setG

Sets green value and return the current color object

meta	description
Returns	<a href="#">Color</a>
Defined in	<a href="#">cocos2d/core/value-types/color.js:282</a>

Parameters

- `green` [Number](#) the new Green component.

## Examples

```
var color = new cc.Color();
color.setG(255); // Color {r: 0, g: 255, b: 0, a: 255}
```

## getB

Gets blue channel value

meta	description
Returns	<a href="#">Number</a>
Defined in	<a href="#">cocos2d/core/value-types/color.js:297</a>

## setB

Sets blue value and return the current color object

meta	description
Returns	<a href="#">Color</a>
Defined in	<a href="#">cocos2d/core/value-types/color.js:306</a>

Parameters

- `blue` [Number](#) the new Blue component.

## Examples

```
var color = new cc.Color();
color.setB(255); // Color {r: 0, g: 0, b: 255, a: 255}
```

### getA

Gets alpha channel value

meta	description
Returns	<a href="#">Number</a>
Defined in	<a href="#">cocos2d/core/value-types/color.js:321</a>

### setA

Sets alpha value and return the current color object

meta	description
Returns	<a href="#">Color</a>
Defined in	<a href="#">cocos2d/core/value-types/color.js:330</a>

## Parameters

- `alpha` [Number](#) the new Alpha component.

## Examples

```
var color = new cc.Color();
color.setA(0); // Color {r: 0, g: 0, b: 0, a: 0}
```

### toCSS

Convert color to css format.

meta	description
Returns	<a href="#">String</a>

meta	description
Defined in	<a href="#">cocos2d/core/value-types/color.js:351</a>

## Parameters

- opt [String](#) "rgba", "rgb", "#rgb" or "#rrggbb".

## Examples

```
```Not found for the example path: temp-
src/engine/docs/utils/api/engine/docs/cocos2d/core/value-types/CCColor/toCSS.js
```

fromHEX

Read hex string and store color data into the current color object, the hex string must be formated as rgba or rgb.

meta	description
Returns	Color
Defined in	cocos2d/core/value-types/color.js:380

Parameters

- hexString [String](#)

Examples

```
var color = cc.Color.BLACK;
color.fromHEX("#FFF33"); // Color {r: 255, g: 255, b: 51, a: 255};
```

toHEX

convert Color to HEX color string. e.g. cc.color(255,6,255) to : "#ff06ff"

meta	description
Returns	String
Defined in	cocos2d/core/value-types/color.js:401

Parameters

- `fmt` [String](#) "#rgb", "#rrggb" or "#rrggbbaa".

Examples

```
var color = cc.Color.BLACK;
color.toHEX("#rgb");      // "000";
color.toHEX("#rrggb");    // "000000";
```

toRGBValue

Convert to 24bit rgb value.

meta	description
Returns	Number
Defined in	cocos2d/core/value-types/color.js:445

Examples

```
var color = cc.Color.YELLOW;
color.toRGBValue(); // 16771844;
```

fromHSV

Read HSV model color and convert to RGB color

meta	description
Returns	Color
Defined in	cocos2d/core/value-types/color.js:458

Parameters

- `h` [Number](#)
- `s` [Number](#)
- `v` [Number](#)

Examples

```
var color = cc.Color.YELLOW;
color.fromHSV(0, 0, 1); // Color {r: 255, g: 255, b: 255, a: 255};
```

toHSV

Transform to HSV model color

meta	description
Returns	Object
Defined in	cocos2d/core/value-types/color.js:536

Examples

```
var color = cc.Color.YELLOW;
color.toHSV(); // Object {h: 0.1533864541832669, s: 0.9843137254901961, v: 1};
```

set

Copies all the properties from another given object to this value.

meta	description
Defined in	cocos2d/core/value-types/value-type.js:84

Parameters

- `source` [ValueType](#) the source to copy

Component Class

Extends [Object](#)

Module: [cc](#)

Base class for everything attached to Node(Entity).

NOTE: Not allowed to use construction parameters for Component's subclasses, because Component is created by the engine.

Index

Properties

- `__eventTargets` Array Register all related EventTargets,...
- `node` Node The node this component is attached to.

- `uuid` String The uuid for editor.
- `_enabled` Boolean
- `enabled` Boolean indicates whether this component is enabled or not.
- `enabledInHierarchy` Boolean indicates whether this component is enabled and its node is also active in the hierarchy.
- `_isOnLoadCalled` Number Returns a value which used to indicate the onLoad get called or not.
- `_name` String
- `_objFlags` Number
- `name` String The name of the object.
- `isValid` Boolean Indicates whether the object is not yet destroyed.

Methods

- `update` This is a lifecycle method.
- `lateUpdate` This is a lifecycle method.
- `__preload` __preload is called before every onLoad.
- `onLoad` When attaching to an active node or its node first activated.
- `start` Called before all scripts' update if the Component is enabled the first time.
- `onEnable` This is a lifecycle method.
- `onDisable` This is a lifecycle method.
- `onDestroy` This is a lifecycle method.
- `onFocusInEditor`
- `onLostFocusInEditor`
- `resetInEditor` Called to initialize the component or node's properties when adding the component the first time or when the Reset command is used.
- `addComponent` Adds a component class to the node.
- `getComponent` Returns the component of supplied type if the node has one attached, null if it doesn't....
- `getComponents` Returns all components of supplied Type in the node.
- `getComponentInChildren` Returns the component of supplied type in any of its children using depth first search.
- `getComponentsInChildren` Returns the components of supplied type in self or any of its children using depth first search.
- `_getLocalBounds` If the component's bounding box is different from the node's, you can implement this method to supply
- `onRestore` for undo/redo operation.
- `schedule` Schedules a custom selector....
- `scheduleOnce` Schedules a callback function that runs only once, with a delay of 0 or larger.
- `unschedule` Unschedules a custom callback function.
- `unscheduleAllCallbacks` unschedule all scheduled callback functions: custom callback functions, and the 'update' callback function....
- `destroy` Actual object destruction will delayed until before rendering.
- `_destruct` Clear all references in the instance.
- `_onPreDestroy` Called before the object being destroyed.
- `_serialize` The customized serialization for this object.
- `_deserialize` Init this object from the custom serialized data.

Details

Properties

__eventTargets

Register all related EventTargets, all event callbacks will be removed in _onPreDestroy

meta	description
Type	Array
Defined in	cocos2d/core/components/CCComponent.js:61

node

The node this component is attached to. A component is always attached to a node.

meta	description
Type	Node
Defined in	cocos2d/core/components/CCComponent.js:75

Examples

```
cc.log(comp.node);
```

uuid

The uuid for editor.

meta	description
Type	String
Defined in	cocos2d/core/components/CCComponent.js:106

Examples

```
cc.log(comp.uuid);
```

_enabled

meta	description
Type	Boolean
Defined in	cocos2d/core/components/CCComponent.js:147

enabled

indicates whether this component is enabled or not.

meta	description
Type	Boolean
Defined in	cocos2d/core/components/CCComponent.js:154

Examples

```
comp.enabled = true;  
cc.log(comp.enabled);
```

enabledInHierarchy

indicates whether this component is enabled and its node is also active in the hierarchy.

meta	description
Type	Boolean
Defined in	cocos2d/core/components/CCComponent.js:185

Examples

```
cc.log(comp.enabledInHierarchy);
```

_isOnLoadCalled

Returns a value which used to indicate the onLoad get called or not.

meta	description
Type	Number
Defined in	cocos2d/core/components/CCComponent.js:201

Examples

```
cc.log(this.\_isOnLoadCalled > 0);
```

`_name`

meta	description
Type	String
Defined in	cocos2d/core/platform/CCObject.js:76

`_objFlags`

meta	description
Type	Number
Defined in	cocos2d/core/platform/CCObject.js:83

`name`

The name of the object.

meta	description
Type	String
Defined in	cocos2d/core/platform/CCObject.js:243

Examples

```
obj.name = "New Obj";
```

isValid

Indicates whether the object is not yet destroyed. (It will not be available after being destroyed)
When an object's `destroy` is called, it is actually destroyed after the end of this frame.
So `isValid` will return false from the next frame, while `isValid` in the current frame will still be true.
If you want to determine whether the current frame has called `destroy`, use `cc.isValid(obj, true)`, but this is often caused by a particular logical requirements, which is not normally required.

meta	description
Type	Boolean
Defined in	cocos2d/core/platform/CCObject.js:261

Examples

```
var node = new cc.Node();
cc.log(node.isValid);      // true
node.destroy();
cc.log(node.isValid);      // true, still valid in this frame
// after a frame...
cc.log(node.isValid);      // false, destroyed in the end of last frame
```

Methods

update

Update is called every frame, if the Component is enabled.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

meta	description
Defined in	cocos2d/core/components/CCComponent.js:222

Parameters

- `dt` [Number](#) the delta time in seconds it took to complete the last frame

lateUpdate

LateUpdate is called every frame, if the Component is enabled.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

meta	description
Defined in	cocos2d/core/components/CCComponent.js:233

__preload

__preload is called before every onLoad. It is used to initialize the builtin components internally, to avoid checking whether onLoad is called before every public method calls. This method should be removed if script priority is supported.

meta	description
Defined in	cocos2d/core/components/CCComponent.js:243

onLoad

When attaching to an active node or its node first activated. onLoad is always called before any start functions, this allows you to order initialization of scripts.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

meta	description
Defined in	cocos2d/core/components/CCComponent.js:254

start

Called before all scripts' update if the Component is enabled the first time. Usually used to initialize some logic which need to be called after all components' onload methods called.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

meta	description
Defined in	cocos2d/core/components/CCComponent.js:267

onEnable

Called when this component becomes enabled and its node is active.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

meta	description
Defined in	cocos2d/core/components/CCComponent.js:280

onDisable

Called when this component becomes disabled or its node becomes inactive.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

meta	description
Defined in	cocos2d/core/components/CCComponent.js:290

onDestroy

Called when this component will be destroyed.

This is a lifecycle method. It may not be implemented in the super class. You can only call its super class method inside it. It should not be called manually elsewhere.

meta	description
Defined in	cocos2d/core/components/CCComponent.js:300

onFocusInEditor

meta	description
Defined in	cocos2d/core/components/CCComponent.js:310

onLostFocusInEditor

meta	description
Defined in	cocos2d/core/components/CCComponent.js:315

resetInEditor

Called to initialize the component or node's properties when adding the component the first time or when the Reset command is used. This function is only called in editor.

meta	description
Defined in	cocos2d/core/components/CCComponent.js:320

addComponent

Adds a component class to the node. You can also add component to node by passing in the name of the script.

meta	description
Returns	Component
Defined in	cocos2d/core/components/CCComponent.js:330

Parameters

- typeOrClassName [Function](#) | [String](#) the constructor or the class name of the component to add

Examples

```
var sprite = node.addComponent(cc.Sprite);
var test = node.addComponent("Test");
```

getComponent

Returns the component of supplied type if the node has one attached, null if it doesn't. You can also get component in the node by passing in the name of the script.

meta	description
Returns	Component
Defined in	cocos2d/core/components/CCComponent.js:348

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
// get sprite component.
var sprite = node.getComponent(cc.Sprite);
// get custom test calss.
var test = node.getComponent("Test");
```

getComponents

Returns all components of supplied Type in the node.

meta	description
Returns	Component[]
Defined in	cocos2d/core/components/CCComponent.js:372

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
var sprites = node.getComponents(cc.Sprite);
var tests = node.getComponents("Test");
```

getComponentInChildren

Returns the component of supplied type in any of its children using depth first search.

meta	description
Returns	Component

meta	description
Defined in	cocos2d/core/components/CCComponent.js:390

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
var sprite = node.getComponentInChildren(cc.Sprite);
var Test = node.getComponentInChildren("Test");
```

getComponentsInChildren

Returns the components of supplied type in self or any of its children using depth first search.

meta	description
Returns	Component[]
Defined in	cocos2d/core/components/CCComponent.js:408

Parameters

- typeOrClassName [Function](#) | [String](#)

Examples

```
var sprites = node.getComponentsInChildren(cc.Sprite);
var tests = node.getComponentsInChildren("Test");
```

_getLocalBounds

If the component's bounding box is different from the node's, you can implement this method to supply a custom axis aligned bounding box (AABB), so the editor's scene view can perform hit test properly.

meta	description
Defined in	cocos2d/core/components/CCComponent.js:428

Parameters

- `out_rect Rect` the Rect to receive the bounding box

onRestore

onRestore is called after the user clicks the Reset item in the Inspector's context menu or performs an undo operation on this component.

If the component contains the "internal state", short for "temporary member variables which not included in its CCClass properties", then you may need to implement this function.

The editor will call the getset accessors of your component to record/restore the component's state for undo/redo operation. However, in extreme cases, it may not work well. Then you should implement this function to manually synchronize your component's "internal states" with its public properties. Once you implement this function, all the getset accessors of your component will not be called when the user performs an undo/redo operation. Which means that only the properties with default value will be recorded or restored by editor.

Similarly, the editor may fail to reset your component correctly in extreme cases. Then if you need to support the reset menu, you should manually synchronize your component's "internal states" with its properties in this function. Once you implement this function, all the getset accessors of your component will not be called during reset operation. Which means that only the properties with default value will be reset by editor.

This function is only called in editor mode.

meta	description
Defined in	cocos2d/core/components/CCComponent.js:441

schedule

Schedules a custom selector.

If the selector is already scheduled, then the interval parameter will be updated without scheduling it again.

meta	description
Defined in	cocos2d/core/components/CCComponent.js:535

Parameters

- `callback function` The callback function
- `interval Number` Tick interval in seconds. 0 means tick every frame.
- `repeat Number` The selector will be executed (repeat + 1) times, you can use cc.macro.REPEAT_FOREVER for tick infinitely.
- `delay Number` The amount of time that the first tick will wait before execution.

Examples

```
var timeCallback = function (dt) {
    cc.log("time: " + dt);
}
this.schedule(timeCallback, 1);
```

scheduleOnce

Schedules a callback function that runs only once, with a delay of 0 or larger.

meta	description
Defined in	cocos2d/core/components/CCComponent.js:572

Parameters

- `callback function` A function wrapped as a selector
- `delay Number` The amount of time that the first tick will wait before execution.

Examples

```
var timeCallback = function (dt) {
    cc.log("time: " + dt);
}
this.scheduleOnce(timeCallback, 2);
```

unschedule

Unschedules a custom callback function.

meta	description
Defined in	cocos2d/core/components/CCComponent.js:589

Parameters

- `callback_fn` [function](#) A function wrapped as a selector

Examples

```
this.unschedule(_callback);
```

unscheduleAllCallbacks

unschedule all scheduled callback functions: custom callback functions, and the 'update' callback function.

Actions are not affected by this method.

meta	description
Defined in	cocos2d/core/components/CCComponent.js:605

Examples

```
this.unscheduleAllCallbacks();
```

destroy

Destroy this Object, and release all its own references to other objects.

Actual object destruction will delayed until before rendering. From the next frame, this object is not usable any more. You can use `cc.isValid(obj)` to check whether the object is destroyed before accessing it.

meta	description
Returns	Boolean
Defined in	cocos2d/core/platform/CCObject.js:296

Examples

```
obj.destroy();
```

_destruct

Clear all references in the instance.

NOTE: this method will not clear the getter or setter functions which defined in the instance of CCObject. You can override the _destruct method if you need, for example: _destruct: function () { for (var key in this) { if (this.hasOwnProperty(key)) { switch (typeof this[key]) { case 'string': this[key] = ""; break; case 'object': case 'function': this[key] = null; break; } } }}

meta	description
Defined in	cocos2d/core/platform/CCObject.js:430

_onPreDestroy

Called before the object being destroyed.

meta	description
Defined in	cocos2d/core/platform/CCObject.js:463

_serialize

The customized serialization for this object. (Editor Only)

meta	description
Returns	object
Defined in	cocos2d/core/platform/CCObject.js:488

Parameters

- exporting [Boolean](#)

_deserialize

Init this object from the custom serialized data.

meta	description
Defined in	cocos2d/core/platform/CCObject.js:498

Parameters

- `data Object` the serialized json data
- `ctx _Deserializer`

Component.EventHandler Class

Module: [cc](#)

Component will register a event to target component's handler. And it will trigger the handler when a certain event occurs.

!@zh “EventHandler” 类用来设置场景中的事件回调，

该类允许用户设置回调目标节点·目标组件名·组件方法名·并可通过 emit 方法调用目标函数。

Index

Properties

- `target Node` Event target
- `component String` Component name
- `handler String` Event handler
- `customEventData String` Custom Event Data

Methods

- `emitEvents`
- `emit` Emit event with params

Details

Properties

target

Event target

meta	description
Type	Node
Defined in	cocos2d/core/components/CCComponentEventHandler.js:51

component

Component name

meta	description
Type	String
Defined in	cocos2d/core/components/CCComponentEventHandler.js:62

handler

Event handler

meta	description
Type	String
Defined in	cocos2d/core/components/CCComponentEventHandler.js:72

customEventData

Custom Event Data

meta	description
Type	String
Defined in	cocos2d/core/components/CCComponentEventHandler.js:83

Methods

emitEvents

meta	description
Defined in	cocos2d/core/components/CCComponentEventHandler.js:96

Parameters

- events [Component.EventHandler\[\]](#)
- params Any

emit

Emit event with params

meta	description
Defined in	cocos2d/core/components/CCComponentEventHandler.js:120

Parameters

- params [Array](#)

Examples

```
// Call Function
var eventHandler = new cc.Component.EventHandler();
eventHandler.target = newTarget;
eventHandler.component = "MainMenu";
eventHandler.handler = "onClick"
eventHandler.emit(["param1", "param2", ....]);
```

constructor Class

Module: [decorator](#) Parent Module: [cc](#)

the device accelerometer reports values for each axis in units of g-force.

Index

Methods

- `setAccelerometerEnabled` whether enable accelerometer event
- `setAccelerometerInterval` set accelerometer interval value

Details

Methods

`setAccelerometerEnabled`

whether enable accelerometer event

meta	description
Defined in	cocos2d/core/platform/CCInputExtension.js:57

Parameters

- `isEnable Boolean`

`setAccelerometerInterval`

set accelerometer interval value

meta	description
Defined in	cocos2d/core/platform/CCInputExtension.js:81

Parameters

- `interval Number`

ContainerStrategy Class

Module: [decorator](#) Parent Module: [cc](#)

cc.game.containerStrategy class is the root strategy class of container's scale strategy, it controls the behavior of how to scale the cc.game.container and cc.game.canvas object

Index

Methods

- `preApply` Manipulation before applying the strategy
- `apply` Function to apply this strategy
- `postApply` Manipulation after applying the strategy

Details

Methods

preApply

Manipulation before applying the strategy

meta	description
Defined in	cocos2d/core/platform/CCView.js:1031

Parameters

- `view` [View](#) The target view

apply

Function to apply this strategy

meta	description
Defined in	cocos2d/core/platform/CCView.js:1041

Parameters

- `view` [View](#)
- `designedResolution` [Size](#)

postApply

Manipulation after applying the strategy

meta	description
Defined in	cocos2d/core/platform/CCView.js:1052

Parameters

- `view` [View](#) The target view

ContentStrategy Class

Module: [decorator](#) Parent Module: [cc](#)

`cc.ContentStrategy` class is the root strategy class of content's scale strategy, it controls the behavior of how to scale the scene and setup the viewport for the game

Index

Methods

- `preApply` Manipulation before applying the strategy
- `apply` Function to apply this strategy
- `postApply` Manipulation after applying the strategy

Details

Methods

preApply

Manipulation before applying the strategy

meta	description
Defined in	cocos2d/core/platform/CCView.js:1137

Parameters

- `view` [View](#) The target view

apply

Function to apply this strategy The return value is {scale: [scaleX, scaleY], viewport: {cc.Rect}}, The target view can then apply these value to itself, it's preferred not to modify directly its private variables

meta	description
Returns	Object
Defined in	cocos2d/core/platform/CCView.js:1147

Parameters

- `view` [View](#)
- `designedResolution` [Size](#)

postApply

Manipulation after applying the strategy

meta	description
Defined in	cocos2d/core/platform/CCView.js:1161

Parameters

- `view` [View](#) The target view

debug Class

Module: [cc](#)

An object to boot the game.

Index

Methods

- `getError` Gets error message with the error id and possible parameters.
- `isDisplayStats` Returns whether or not to display the FPS informations.
- `setDisplayStats` Sets whether display the FPS on the bottom-left corner.

Details

Methods

getError

Gets error message with the error id and possible parameters.

meta	description
Returns	String
Defined in	cocos2d/core/CCDebug.js:343

Parameters

- `errorId` [Id](#)
- `param` [Any](#)

isDisplayStats

Returns whether or not to display the FPS informations.

meta	description
Returns	Boolean
Defined in	cocos2d/core/CCDebug.js:353

setDisplayStats

Sets whether display the FPS on the bottom-left corner.

meta	description
Defined in	cocos2d/core/CCDebug.js:363

Parameters

- `displayStats` [Boolean](#)

Details Class

Module: [decorator](#) Parent Module: [cc](#)

Contains information collected during deserialization

Index

Properties

- `uuidList` String[] list of the depends assets' uuid
- `uuidObjList` Object[] the obj list whose field needs to load asset by uuid
- `uuidPropList` String[] the corresponding field name which referenced to the asset

Methods

- `reset`
- `push`

Details

Properties

uuidList

list of the depends assets' uuid

meta	description
Type	String[]
Defined in	cocos2d/core/platform/deserialize.js:44

uuidObjList

the obj list whose field needs to load asset by uuid

meta	description
Type	Object[]

meta	description
Defined in	cocos2d/core/platform/deserialize.js:49

uuidPropList

the corresponding field name which referenced to the asset

meta	description
Type	String[]
Defined in	cocos2d/core/platform/deserialize.js:54

Methods

reset

meta	description
Defined in	cocos2d/core/platform/deserialize.js:63

push

meta	description
Defined in	cocos2d/core/platform/deserialize.js:97

Parameters

- obj [Object](#)
- propName [String](#)
- uuid [String](#)

