JavaScript™ for Acrobat® 3D Annotations
API Reference
Contents

Preface .................................................................................................................................................. 8
  What’s in this guide? .......................................................................................................................... 8
  Who should read this guide? .............................................................................................................. 8
  Related documentation .................................................................................................................... 8

1 Introduction ........................................................................................................................................ 9
  Object overview ............................................................................................................................... 11
    Basic objects ................................................................................................................................. 11
    Scene object ................................................................................................................................. 11
    Canvas object .............................................................................................................................. 11
    Runtime object ........................................................................................................................... 12
    Console object ............................................................................................................................ 12
    Resource objects ......................................................................................................................... 12
    Event handlers ............................................................................................................................ 12
    CamaraEvent ............................................................................................................................. 12
    KeyEvent ....................................................................................................................................... 13
    MenuEvent ................................................................................................................................... 13
    MouseEvent ................................................................................................................................ 13
    RenderEvent ............................................................................................................................... 13
    ScrollWheelEvent ...................................................................................................................... 14
    SelectionEvent ........................................................................................................................... 14
    TimeEvent .................................................................................................................................. 14
    ToolEvent ................................................................................................................................... 14

2 JavaScript Objects for Acrobat 3D ............................................................................................... 15
  Animation ....................................................................................................................................... 16
  Background .................................................................................................................................... 17
    getColor ....................................................................................................................................... 17
    getImage ...................................................................................................................................... 17
    setColor .................................................................................................................................... 17
    setImage ..................................................................................................................................... 18
  Bone ............................................................................................................................................... 19
  BoundingBox ................................................................................................................................... 20
  Camera .......................................................................................................................................... 21
    getScreenFromPosition ............................................................................................................... 22
    getDirectionFromScreen ............................................................................................................ 23
  CamaraEvent ................................................................................................................................ 24
  CameraEventHandler .................................................................................................................... 25
    CameraEventHandler .................................................................................................................. 25
    onEvent ..................................................................................................................................... 25
  Canvas ........................................................................................................................................... 26
    getCamera .................................................................................................................................. 26
    setCamera .................................................................................................................................. 26
  ClippingPlane ............................................................................................................................... 27
    remove ....................................................................................................................................... 27
  Color ............................................................................................................................................. 28
    Color ......................................................................................................................................... 28
rotateAboutY .................................................................56
rotateAboutYInPlace .................................................56
rotateAboutZ ..............................................................56
rotateAboutZInPlace ................................................57
scale ..............................................................................57
scaleInPlace ...............................................................58
set ...................................................................................58
set ...................................................................................58
set ...................................................................................59
setIdentity .................................................................59
setView .........................................................................59
transformDirection ....................................................60
transformPosition .......................................................60
translate ........................................................................60
translateInPlace ........................................................61
transposeInPlace .......................................................61
MenuEvent ....................................................................62
MenuEventHandler ......................................................63
  MenuEventHandler ......................................................63
onEvent .........................................................................63
Mesh ...........................................................................64
  computeBoundingBox ..................................................64
MouseEvent ....................................................................65
  MouseEventHandler ....................................................67
    MouseEventHandler ..................................................67
onEvent .........................................................................68
Node ...........................................................................69
  detachFromCurrentAnimation ....................................70
Procedural ....................................................................71
  Quaternion ..................................................................72
    Quaternion ..................................................................72
    Quaternion ..................................................................72
    Quaternion ..................................................................72
    interpolate ..............................................................73
    interpolateInPlace ..................................................73
    normalize ..................................................................73
RenderEvent ...............................................................75
RenderEventHandler .....................................................76
  RenderEventHandler ....................................................76
onEvent .........................................................................76
RenderOptions ............................................................77
Resource .......................................................................79
  Resource .....................................................................79
Runtime .........................................................................80
  addCustomMenuItem ..................................................83
  addCustomToolButton ...............................................83
  addEventHandler .......................................................84
  disableTool .............................................................84
  enableTool ...............................................................84
  getEventHandler .......................................................85
  getRendererName ......................................................85
<table>
<thead>
<tr>
<th>Function/METHOD</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scene</td>
<td>91</td>
</tr>
<tr>
<td>activateAnimation</td>
<td>91</td>
</tr>
<tr>
<td>addFlashForeground</td>
<td>97</td>
</tr>
<tr>
<td>addModel</td>
<td>98</td>
</tr>
<tr>
<td>createClippingPlane</td>
<td>98</td>
</tr>
<tr>
<td>createLight</td>
<td>98</td>
</tr>
<tr>
<td>createSquareMesh</td>
<td>98</td>
</tr>
<tr>
<td>computeBoundingBox</td>
<td>99</td>
</tr>
<tr>
<td>update</td>
<td>99</td>
</tr>
<tr>
<td>SceneObject</td>
<td>100</td>
</tr>
<tr>
<td>SceneObjectList</td>
<td>101</td>
</tr>
<tr>
<td>getByGUID</td>
<td>101</td>
</tr>
<tr>
<td>byID</td>
<td>101</td>
</tr>
<tr>
<td>byIndex</td>
<td>101</td>
</tr>
<tr>
<td>byName</td>
<td>102</td>
</tr>
<tr>
<td>removeAll</td>
<td>102</td>
</tr>
<tr>
<td>removeByIndex</td>
<td>102</td>
</tr>
<tr>
<td>removeItem</td>
<td>103</td>
</tr>
<tr>
<td>ScrollWheelEvent</td>
<td>104</td>
</tr>
<tr>
<td>ScrollWheelEventHandler</td>
<td>105</td>
</tr>
<tr>
<td>onEvent</td>
<td>105</td>
</tr>
<tr>
<td>SelectionEvent</td>
<td>106</td>
</tr>
<tr>
<td>SelectionEventHandler</td>
<td>107</td>
</tr>
<tr>
<td>onEvent</td>
<td>107</td>
</tr>
<tr>
<td>StateEvent</td>
<td>108</td>
</tr>
<tr>
<td>StateEventHandler</td>
<td>109</td>
</tr>
<tr>
<td>onEvent</td>
<td>109</td>
</tr>
<tr>
<td>Syntax</td>
<td>109</td>
</tr>
<tr>
<td>Texture</td>
<td>110</td>
</tr>
<tr>
<td>getImage</td>
<td>110</td>
</tr>
<tr>
<td>setimage</td>
<td>111</td>
</tr>
<tr>
<td>TimeEvent</td>
<td>112</td>
</tr>
<tr>
<td>TimeEventHandler</td>
<td>113</td>
</tr>
<tr>
<td>onEvent</td>
<td>113</td>
</tr>
</tbody>
</table>
3 New Features and Changes......................................................................................................................... 124

Acrobat 9.0 changes................................................................................................................................. 124
Acrobat 8.1 changes................................................................................................................................. 125
Acrobat 8.0 changes................................................................................................................................. 126

Index ........................................................................................................................................................................ 127
Preface

The JavaScript™ API lets you manipulate 3D annotations within Adobe® PDF documents.

What’s in this guide?

This document provides a brief overview of the API followed by a description of the objects.

Who should read this guide?

This guide is for developers who want to enhance the 3D experience of the user beyond the default behaviors. Using the JavaScript API for 3D annotations, you can specify the render modes and 3D matrix transformations of any of the individual meshes; set camera position, target, and field of view; detect mouse and keyboard events; control animations; and many more behaviors.

Related documentation

This document refers to the following sources for additional information about 3D annotations, JavaScript, and related technologies. The Adobe Acrobat® documentation is available through the Acrobat Developer Center.

<table>
<thead>
<tr>
<th>Document</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developing Acrobat Applications</td>
<td>Using JavaScript to develop and enhance standard workflows in Acrobat and Acrobat Reader®.</td>
</tr>
<tr>
<td>Using JavaScript</td>
<td></td>
</tr>
<tr>
<td>JavaScript for Acrobat API Reference</td>
<td>Detailed descriptions of JavaScript APIs for developing and enhancing workflows in Acrobat and Acrobat Reader.</td>
</tr>
<tr>
<td>PDF Reference</td>
<td>A detailed description of the PDF file format.</td>
</tr>
</tbody>
</table>
Introduction

To create 3D annotations and to attach scripts to them using this API, you need Adobe® Acrobat® Professional. Scripts attached to 3D annotations can run on Acrobat Pro Extended, Acrobat Pro DC, Acrobat Standard DC, and Acrobat Reader® for Windows® and Mac OS® platforms. Unless otherwise noted, all JavaScript objects, properties, and methods have support starting in version 7.0.

The 3D JavaScript engine, which is distinct from the JavaScript engine for Acrobat, can be accessed in one of two ways. The primary way is by attaching a default script to the 3D annotation. This can be accomplished while placing a 3D annotation using the 3D Tool or on an existing 3D annotation by accessing its properties dialog box using the Select Object tool. This script will be run directly by the 3D JavaScript engine.

In addition, Acrobat provides a mechanism to directly access the entire 3D JavaScript engine API from within the Acrobat scripting engine by means of the JavaScript `Annot3D.context3D` property. For more information about JavaScript for Acrobat and its `Annot3D` object, see the JavaScript for Acrobat API Reference and Developing Acrobat Applications Using JavaScript.

The following example illustrates how to access the 3D JavaScript engine. In this example, a button (or link) contains JavaScript code that rotates the U3D object named "Axes".

```javascript
// Get index of page containing the Annot3D object (count starts at 0).
pageIndex = this.pageNum;

// Index of the Annot3D (count starts at 0).
annotIndex = 0;

// Get a reference to the Annot3D script context.
c3d = this.getAnnots3D( pageIndex )[ annotIndex ].context3D;

// Get a reference to the node in the scene named "Axes".
axes = c3d.scene.nodes.getByName( "Axes" );

// Rotate the object about the X-Axis PI/6 radians (30 degrees).
axes.transform.rotateAboutXInPlace( Math.PI / 6 );
```

More extensive actions can be executed by having a button or link get the `SceneContext3d` object and call a function defined in the default script of the 3D annotation, as in the following example.

```javascript
// Get the Annot3D script context of the targeted annotation.
context3D = getAnnots3D(0)[0].context3D;

// Call the JavaScript function setRenderMode() defined in the default script of the referenced 3D annotation.
context3D.setRenderMode("transparent");
```

The default script of the 3D annotation makes the definition.

```javascript
function setRenderMode( renderModeName ) {
    for (var i=0; i < scene.meshes.count; i++) {
```
```javascript
scene.meshes.getByIndex(i).renderMode = renderModeName;
```
Object overview

This section provides an overview of the objects in the 3D JavaScript API.

Basic objects

There are several basic objects, such as Color, Matrix4x4, and Vector3, that are used to create general-purpose objects. The basic objects are used throughout the API and are only meaningful when attached to objects such as Scene or Runtime. For example, you could create a Color object and use it to set the Background color of a Canvas.

**Vector3 Examples**

```javascript
v1 = new Vector3( 1.2, 3, 4.5 );
v2 = new Vector3( 5, 8, 13 );
v3 = new Vector3();
```

**Matrix4x4 Examples**

```javascript
m1 = new Matrix4x4().rotateAboutX(Math.PI/1.5).rotateAboutY(Math.PI/3);
m2 = new Matrix4x4().rotateAboutZ(Math.PI/4).translate(new Vector3(0,5,0));
m3 = new Matrix4x44x44(m1);
```

**Color Examples**

```javascript
c1 = new Color( 0.6, 0.8, 1.0 ); // light blue
c2 = new Color( 0.5, 0.5, 0.5 ); // middle grey
c3 = new Color(); // black
```

```javascript
// A function to blend two Colors
Color.prototype.blend = function( color, amount )
{
    red = ( this.r * ( 1 - amount ) ) + ( color.r * amount );
    green = ( this.g * ( 1 - amount ) ) + ( color.g * amount );
    blue = ( this.b * ( 1 - amount ) ) + ( color.b * amount );
    return( new Color( red, green, blue ) );
}
c4 = c1.blend( c2, 0.25 );
```

Scene object

The Scene is an object that contains all of the 3D-related content. It can be accessed using the global variable scene, which is a reference to the main Scene object. Most of the contents of the Scene are structured into a hierarchy of Node objects, and maintains lists of all these objects in the form of a SceneObjectList.

For more information, see [Scene](#).

Canvas object

Represents a rectangular region into which a Scene is rendered from a particular viewpoint.

For more information, see [Canvas](#).
Runtime object

The Runtime object is used to represent the instance of the playback engine. It manages all event processing and places where the graphic and textual content is rendered. It is accessed via the global variable runtime, which is a reference to the main Runtime object.

For more information, see Runtime.

Console object

The Console is the Acrobat text output area. It is helpful in debugging scripts.

Resource objects

Some objects, such as Image, are driven by content that is streamed from a file or over a network. To create an Image, load a .png, .jpg, or .gif file as a Resource, which you may subsequently use to create a new Image object, as shown in the following example:

```javascript
faceRes = new Resource("pdf://picture.jpg");
faceImage = new Image( faceRes );
aMaterial = scene.meshes.getByIndex(0).material;
aMaterial.diffuseTexture.setImage( faceImage );
```

For more information, see Resource and Image.

Event handlers

There are several types of event handlers:

- CameraEventHandler
- KeyEventHandler
- MouseEventHandler
- MenuEventHandler
- RenderEventHandler
- ScrollWheelEventHandler
- SelectionEventHandler
- TimeEventHandler
- ToolEventHandler

Each one responds to a different type of event during simulation. They use a callback mechanism to run a function when an event occurs. The event is passed as an argument to the event handler's `onEvent` function so that it can be queried when the function runs. Event handlers are registered via the `addEventHandler` method, of the Runtime object.

CamaraEvent

A CamaraEvent is created when a View is selected.

For more information, see CameraEvent.
KeyEvent

A KeyEvent is created when a key is pressed or released while the 3D Canvas is in focus. The following example illustrates how to handle a key event:

```javascript
myKeyHandler = new KeyEventHandler();
myKeyHandler.onEvent = function( event )
{
    console.print( "Key pressed with code: " + event.characterCode );
}
runtime.addEventHandler( myKeyHandler );
```

For more information, see KeyEvent.

MenuEvent

A MenuEvent is created when a custom menu item is selected. To create a custom menu item on the context menu, invoke the Runtime object's addCustomMenuItem method, which allows a script to be attached to the item selection event.

For more information, see MenuEvent.

MouseEvent

A MouseEvent is created when the mouse is clicked on an active 3D Canvas or the cursor moves over an active 3D Canvas. The following syntax could be used to handle a mouse event:

```javascript
myMouseHandler = new MouseEventHandler();
myMouseHandler.onMouseDown = true;
myMouseHandler.target = scene.meshes.getByIndex(0);
myMouseHandler.onEvent = function( event )
{
    console.print( "Mouse down at pixel " + event.mouseX );
    console.print( ", " + event.mouseY );
}
runtime.addEventHandler( myMouseHandler );
```

For more information, see MouseEvent.

RenderEvent

A RenderEvent is created immediately before an instance of the Canvas is drawn. If there is a split view in Acrobat resulting in two visible 3D rendered areas, a unique RenderEvent will be called for each of them. This is necessary in the case of a camera-aligned image (sprite) in the 3D content that needs to be pixel-aligned. Since the pixel dimensions of the two areas are possibly different, there are two callbacks that pass the different dimensions. This makes it possible to modify the Scene in the appropriate manner before it is drawn.

For more information, see RenderEvent.
ScrollWheelEvent

A `ScrollWheelEvent` object is created when the mouse scroll wheel is activated over an active 3D Canvas object.

For more information, see `ScrollWheelEvent`.

SelectionEvent

A `SelectionEvent` object is created when an object is selected from an active 3D Canvas object or from a model tree. If the selection is made from a Canvas object, a `MouseEvent` is also created.

For more information, see `SelectionEvent`.

TimeEvent

A `TimeEvent` is created when the 3D annotation is enabled and simulation is active. The time and deltaTime properties are measured in terms of simulation time, not real time. `TimeEvent` objects are used to drive animation. If you need an accurate, real-time measurement, use the JavaScript Date object. The following syntax is used to handle a time event:

```javascript
myTimeHandler = new TimeEventHandler();
myTimeHandler.onEvent = function( event )
{
    console.print( "Current simulation time is:" + event.time );
    console.print( " second(s)" );
}
runtime.addEventHandler( myTimeHandler );
```

For more information, see `TimeEvent`.

ToolEvent

A `ToolEvent` is created when a tool is clicked in the Acrobat 3D toolbar. The `Runtime` object’s `addCustomToolButton` method allows you to add a custom tool to the toolbar which will also be generated, and allows a script to be attached to the tool selection event.

For more information, see `ToolEvent`.
JavaScript Objects for Acrobat 3D

This chapter describes the following 3D JavaScript objects:

- Animation
- Background
- Bone
- BoundingBox
- Camera
- CameraEvent
- CameraEventHandler
- Canvas
- ClippingPlane
- Color
- Console
- Dummy
- FlashEvent
- FlashEventHandler
- FlashMovie
- HitInfo
- Host
- Image
- KeyEvent
- KeyEventHandler
- Light
- Material
- Matrix4x4
- MenuEvent
- MenuEventHandler
- Mesh
- MouseEvent
- MouseEventHandler
- Node
- Procedural
- Quaternion
- RenderEvent
- RenderEventHandler
- RenderOptions
- Resource
- Runtime
- Scene
- SceneObject
- SceneObjectList
- ScrollWheelEvent
- ScrollWheelEventHandler
- SelectionEvent
- SelectionEventHandler
- StateEvent
- StateEventHandler
- Texture
- TimeEvent
- TimeEventHandler
- ToolEvent
- ToolEventHandler
- Vector3
- View

Note: A property labeled as R (read-only) is one whose value cannot be set. An object labeled as R (read-only) is one whose reference cannot be modified, though the object itself can be set and its properties may be modified. Unless otherwise indicated, all properties and objects labeled with R/W have read/write access.
Animation

A type of SceneObject, used to store keyframe animation sequences of Node objects in the Scene. In addition to the methods and properties below, it also contains the same methods and properties as a SceneObject.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>currentTime</td>
<td>number</td>
<td>R/W</td>
<td>The current time measured in seconds.</td>
</tr>
<tr>
<td>endTime</td>
<td>number</td>
<td>R</td>
<td>The end time of the sequence, measured in seconds.</td>
</tr>
<tr>
<td>framesPerSecond</td>
<td>number</td>
<td>R</td>
<td>The number of frames per second used to author the sequence.</td>
</tr>
<tr>
<td>length</td>
<td>number</td>
<td>R</td>
<td>The length of the Animation, measured in seconds.</td>
</tr>
<tr>
<td>startTime</td>
<td>number</td>
<td>R</td>
<td>The start time of the sequence, measured in seconds.</td>
</tr>
</tbody>
</table>
Background

Represents the background of a Canvas. It can be used as a target of a MouseEventHandler. (See Canvas and MouseEventHandler.)

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>image</td>
<td>Image</td>
<td>R/W</td>
<td>Acrobat 7.0.7&lt;br&gt;The Image to be used by the Background.</td>
</tr>
<tr>
<td>FlashMovie</td>
<td>FlashMovie</td>
<td>R/W</td>
<td>Acrobat 9.0&lt;br&gt;The FlashMovie to be used by the Background.&lt;br&gt;FlashMovie replaces any Image or Color currently being used by the Background</td>
</tr>
</tbody>
</table>

getColor

Obtains the background Color.

Syntax

g getColor()  

Returns

A Color object representing the background color of the Canvas.

getImage

Deprecated

Obtains the background Image.

Syntax

g getImage()  

Returns

An Image object representing the background image of the Canvas.

setColor

Sets the background Color. If only one color is passed to this method, the background is a constant color. If two colors are passed to this method, the background is a linear gradient from top to bottom, with the first color argument representing the top color and the second representing the bottom color.
Syntax

```javascript
setColor(topColor, bottomColor)
```

Parameters

<table>
<thead>
<tr>
<th>topColor</th>
<th>A Color object representing the desired background color. If <code>bottomColor</code> is used, <code>topColor</code> represents the top background color used in a linear gradient.</th>
</tr>
</thead>
<tbody>
<tr>
<td>bottomColor</td>
<td>(Optional) A Color object representing the bottom background color used in a linear gradient.</td>
</tr>
</tbody>
</table>

Returns

`undefined`

**setImage**

Deprecated

Sets the background Image.

Syntax

```javascript
setImage(image)
```

Parameters

| image | An Image object representing the desired background image. |

Returns

`undefined`
Bone

A type of Node used to modify the shape of a Mesh, and is usually moved over time to create animated characters. It contains the same methods and properties as a Node.

Related objects are Node and Mesh.
BoundingBox

Represents an axis-aligned bounding box.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>center</td>
<td>Vector3</td>
<td>R</td>
<td>Acrobat 7.0.7  The coordinates of the BoundingBox center.</td>
</tr>
<tr>
<td>max</td>
<td>Vector3</td>
<td>R</td>
<td>The coordinates of the BoundingBox corner with the greatest x, y, and z values.</td>
</tr>
<tr>
<td>min</td>
<td>Vector3</td>
<td>R</td>
<td>The coordinates of the BoundingBox corner with the smallest x, y, and z values.</td>
</tr>
</tbody>
</table>
Camera

A Node that controls the projection from world space to screen space. In addition to the methods and properties below, it also contains the same methods and properties as a Node.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
</table>
| absoluteBindingScale | number     | R/W    | Acrobat 7.0 
The absolute binding scale value for the camera.                     |
| binding              | string     | R/W    | Acrobat 7.0 
The view plane calculation type, which can take one of the following values: |
|                      |            |        | ● "min"                                                                    |
|                      |            |        | ● "max"                                                                    |
|                      |            |        | ● "horizontal"                                                             |
|                      |            |        | ● "vertical"                                                               |
| BINDING_HORIZONTAL   | string     | R      | Acrobat 7.0.7 
A string constant for the binding value of "horizontal".                   |
| BINDING_MAX          | string     | R      | Acrobat 7.0.7 
A string constant for the binding value of "max".                           |
| BINDING_MIN          | string     | R      | Acrobat 7.0.7 
A string constant for the binding value of "min".                           |
| BINDING_VERTICAL     | string     | R      | Acrobat 7.0.7 
A string constant for the binding value of "vertical".                       |
| far                  | number     | R/W    | Acrobat 7.0 
The distance from the Camera to the far clipping plane. A value of -1 for both near and far signifies to use auto clipping plane calculations. |
| fov                  | number     | R/W    | Acrobat 7.0 
The size of the field of view for perspective Camera objects, measured in radians. |
| near                 | number     | R/W    | Acrobat 7.0 
The distance from the Camera to the near clipping plane. A value of -1 for both near and far signifies to use auto clipping plane calculations. |
| position             | Vector3    | R      | Acrobat 7.0 
The position of the origin of the Camera in world space.            |
| positionLocal        | Vector3    | R      | Acrobat 7.0 
The position of the origin of the Camera in local space.           |
**getScreenFromPosition**

Obtains the screen coordinates of the provided 3D position.

**Syntax**

```javascript
getScreenFromPosition(position, canvasWidth, canvasHeight)
```

**Parameters**

- **position**
  A `Vector3` object representing the 3D position.

---

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>projectionType</code></td>
<td>string</td>
<td>R/W</td>
<td>The type of projection, which can take one of the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- &quot;perspective&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- &quot;orthographic&quot;</td>
</tr>
<tr>
<td><code>roll</code></td>
<td>number</td>
<td>R/W</td>
<td>The roll angle of the Camera, measured in radians.</td>
</tr>
<tr>
<td><code>target</code></td>
<td>Node</td>
<td>R</td>
<td>The current Node used as the Camera object’s target.</td>
</tr>
<tr>
<td><code>targetPosition</code></td>
<td>Vector3</td>
<td>R</td>
<td>The position of the Camera object’s target in world space.</td>
</tr>
<tr>
<td><code>targetPositionLocal</code></td>
<td>Vector3</td>
<td>R/W</td>
<td>The position of the Camera object’s target in local space.</td>
</tr>
<tr>
<td><code>TYPE_ORTHOGRAPHIC</code></td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the camera projection type of &quot;orthographic&quot;.</td>
</tr>
<tr>
<td><code>TYPE_PERSPECTIVE</code></td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the camera projection type of &quot;perspective&quot;.</td>
</tr>
<tr>
<td><code>up</code></td>
<td>Vector3</td>
<td>R</td>
<td>The up direction in world space.</td>
</tr>
<tr>
<td><code>upLocal</code></td>
<td>Vector3</td>
<td>R</td>
<td>The up direction in local space.</td>
</tr>
<tr>
<td><code>useAbsoluteBinding</code></td>
<td>Boolean</td>
<td>R</td>
<td>Acrobat 7.0 Determines whether the camera uses absolute binding for its projection.</td>
</tr>
<tr>
<td><code>viewPlaneSize</code></td>
<td>number</td>
<td>R/W</td>
<td>The size of the view plane for orthographic Camera objects, measured in scene units.</td>
</tr>
</tbody>
</table>
getDirectionFromScreen

Obtains the direction from the normalized coordinates.

**Syntax**

```javascript
getDirectionFromScreen(x, y, canvasWidth, canvasHeight)
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>x</code></td>
<td>The x-coordinate, measured in pixels.</td>
</tr>
<tr>
<td><code>y</code></td>
<td>The y-coordinate, measured in pixels.</td>
</tr>
<tr>
<td><code>canvasWidth</code></td>
<td>The width of the Canvas, measured in pixels.</td>
</tr>
<tr>
<td><code>canvasHeight</code></td>
<td>The height of the Canvas, measured in pixels.</td>
</tr>
</tbody>
</table>

**Returns**

A `Vector3` object representing the direction.

See `Vector3` for more information on the return object.
CameraEvent

Describes the format of the object that is passed as an argument to the `onEvent` method of the `CameraEventHandler` object.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>binding</td>
<td>string</td>
<td>R</td>
<td>The view plane calculation type, which can take one of the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;min&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;max&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;horizontal&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;vertical&quot;</td>
</tr>
<tr>
<td>canvas</td>
<td>Canvas</td>
<td>R</td>
<td>The Canvas in which the event took place.</td>
</tr>
<tr>
<td>currentTool</td>
<td>string</td>
<td>R</td>
<td>The name of the current tool.</td>
</tr>
<tr>
<td>far</td>
<td>number</td>
<td>R</td>
<td>The distance from the Camera to the far clipping plane. A value of -1 for both near and far signifies to use auto clipping plane calculations.</td>
</tr>
<tr>
<td>fov</td>
<td>number</td>
<td>R</td>
<td>The size of the field of view for perspective Camera objects, measured in radians.</td>
</tr>
<tr>
<td>isNewCanvas</td>
<td>Boolean</td>
<td>R</td>
<td>Deprecated</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Determines whether this is the first event for this Canvas.</td>
</tr>
<tr>
<td>near</td>
<td>number</td>
<td>R</td>
<td>The distance from the Camera to the near clipping plane. A value of -1 for both near and far signifies to use auto clipping plane calculations.</td>
</tr>
<tr>
<td>projectionType</td>
<td>string</td>
<td>R</td>
<td>The type of projection, which can take one of the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;perspective&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;orthographic&quot;</td>
</tr>
<tr>
<td>targetDistance</td>
<td>number</td>
<td>R</td>
<td>The distance from the Camera to its target.</td>
</tr>
<tr>
<td>transform</td>
<td>Matrix4x4</td>
<td>R</td>
<td>The Camera object’s transformation matrix.</td>
</tr>
<tr>
<td>view</td>
<td>View object</td>
<td>R</td>
<td>The name of the view being activated.</td>
</tr>
<tr>
<td>viewPlaneSize</td>
<td>number</td>
<td>R</td>
<td>The size of the view plane, measured in scene units.</td>
</tr>
</tbody>
</table>
CameraEventHandler

Exposes a callback mechanism that allows a function to be evaluated when an camera event occurs. Event handlers are registered with the Runtime `addEventHandler` method.

**CameraEventHandler**

A constructor that creates a new `CameraEventHandler` object.

**Syntax**

```javascript
new CameraEventHandler()
```

**Returns**

A `CameraEventHandler` object.

**onEvent**

A method that is called when a view is selected from the list of views on the 3D toolbar or in the context menu for an active 3D annotation.

**Syntax**

```javascript
onEvent(event)
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>event</td>
<td>A <code>CameraEvent</code> object representing the event.</td>
</tr>
</tbody>
</table>

**Returns**

`undefined`
Canvas

Represents a rectangular region into which the Scene is rendered from the viewpoint of the attached Camera.

See related objects, Scene and Camera.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>background</td>
<td>Background</td>
<td>R</td>
<td>The Background object associated with the Canvas.</td>
</tr>
</tbody>
</table>

getCamera

Obtains the Camera object attached to the Canvas.

Syntax

getCamera()

Returns

A Camera object.

setCamera

Sets the Camera object attached to the Canvas.

Syntax

setCamera(camera)

Parameters

| camera    | The Camera object used to set the object’s value. |

Returns

undefined
ClippingPlane

An object representing a plane, within the Scene, that clips all geometry on one side of it. It is created by invoking the `createClippingPlane` method of the Scene object.

**remove**

Removes the ClippingPlane object from the Scene.

**Syntax**

```javascript
remove()
```

**Returns**

undefined
Color

An object that represents a RGB encoded color.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>number</td>
<td>The blue component, which can be a value from 0.0 to 1.0.</td>
</tr>
<tr>
<td>g</td>
<td>number</td>
<td>The green component, which can be a value from 0.0 to 1.0.</td>
</tr>
<tr>
<td>r</td>
<td>number</td>
<td>The red component, which can be a value from 0.0 to 1.0.</td>
</tr>
</tbody>
</table>

Color

A constructor that creates a Color object, initialized to black.

Syntax

new Color()

Returns

A Color object, initialized to black.

Color

A constructor that creates a Color object, initialized to the supplied RGB values.

Syntax

new Color(r, g, b)

Parameters

<table>
<thead>
<tr>
<th>r</th>
<th>The red component, which can be a value from 0.0 to 1.0.</th>
</tr>
</thead>
<tbody>
<tr>
<td>g</td>
<td>The green component, which can be a value from 0.0 to 1.0.</td>
</tr>
<tr>
<td>b</td>
<td>The blue component, which can be a value from 0.0 to 1.0.</td>
</tr>
</tbody>
</table>

Returns

A Color object, initialized to the supplied RGB values.

set

Sets the Color object's value using an existing Color object
Syntax
  set(color)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>color</td>
<td>The <code>Color</code> object used to set the object’s value.</td>
</tr>
</tbody>
</table>

Returns

undefined

set

Acrobat 7.0.7

Sets the `Color` object’s value using the given RGB components.

Syntax
  set(r, g, b)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>The red component, which can be a value from 0.0 to 1.0.</td>
</tr>
<tr>
<td>g</td>
<td>The green component, which can be a value from 0.0 to 1.0.</td>
</tr>
<tr>
<td>b</td>
<td>The blue component, which can be a value from 0.0 to 1.0.</td>
</tr>
</tbody>
</table>

Returns

undefined

set3

Deprecated

Sets the `Color` object’s value using the given RGB components.

Syntax
  set3(r, g, b)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>The red component, which can be a value from 0.0 to 1.0.</td>
</tr>
<tr>
<td>g</td>
<td>The green component, which can be a value from 0.0 to 1.0.</td>
</tr>
<tr>
<td>b</td>
<td>The blue component, which can be a value from 0.0 to 1.0.</td>
</tr>
</tbody>
</table>
Returns

undefined
**Console**

This object can direct output to the JavaScript console in Acrobat for debugging purposes. The variable `console` is a global reference to this object.

**print**

Prints a string to the JavaScript Console.

**Syntax**

```javascript
print(string)
```

**Parameters**

| string       | The text to be printed to the JavaScript Console. |

**Returns**

undefined

**println**

Prints a string with an accompanying newline to the JavaScript Console.

**Syntax**

```javascript
println(string)
```

**Parameters**

| string       | The text to be printed to the JavaScript Console. |

**Returns**

undefined
Dummy

Deprecated

A Node object used as an empty placeholder or a group within a Scene.
FlashEvent

Acrobat 9.0

An object that is passed as an argument to the `onEvent` method of the FlashEventHandler object.

## Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>command</td>
<td>string</td>
<td>R</td>
<td>For a <code>FlashEvent</code> of type &quot;command&quot;, this is the string representation of the command that has been sent through the ActionScript <code>FSCommand</code> function or through the <code>ExternalInterface.call</code> method. To execute the command, run the JavaScript function <code>eval</code> with the command string as an argument.</td>
</tr>
<tr>
<td>target</td>
<td>FlashMovie</td>
<td>R</td>
<td>The target FlashMovie that the <code>FlashEvent</code> is from.</td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td>R</td>
<td>The type of <code>FlashEvent</code>, which can be &quot;command&quot;, &quot;progress&quot;, or &quot;stateChange&quot;.</td>
</tr>
<tr>
<td>TYPE_COMMAND</td>
<td>string</td>
<td>R</td>
<td>A string constant for the <code>FlashEvent</code> type of &quot;command&quot;.</td>
</tr>
<tr>
<td>TYPE_PROGRESS</td>
<td>string</td>
<td>R</td>
<td>A string constant for the <code>FlashEvent</code> type of &quot;progress&quot;.</td>
</tr>
<tr>
<td>TYPE_STATECHANGE</td>
<td>string</td>
<td>R</td>
<td>A string constant for the <code>FlashEvent</code> type of &quot;stateChange&quot;.</td>
</tr>
<tr>
<td>value</td>
<td>integer</td>
<td>R</td>
<td>The value for the corresponding type of <code>FlashEvent</code>. The interpretation of value depends on the event type, &quot;progress&quot; or &quot;stateChange&quot;.</td>
</tr>
</tbody>
</table>

  - "progress": `value` is an integer from 0 to 100 representing the load progress of the FlashMovie.
  - "stateChange": `value` is an integer signifying the ready state of the FlashMovie. Permitted values are 0 (Loading), 1 (Uninitialized), 2 (Loaded), 3 (Interactive), 4 (Complete).
FlashEventHandler

Acrobat 9.0

An object that exposes a callback mechanism that allows a function to be evaluated when an event occurs in a FlashMovie object. Event handlers are registered with the Runtime `addEventHandler` method.

**Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>target</td>
<td>FlashMovie</td>
<td>R/W</td>
<td>When set, the FlashEventHandler will only report FlashEvents from the provided target FlashMovie.</td>
</tr>
</tbody>
</table>

**onEvent**

A method that is called when an `ExternalInterface.call` method or `MMExecute` command is invoked from the FlashMovie's ActionScript.

**Syntax**

```
onEvent(event)
```

**Parameters**

- **event** A FlashEvent object representing the event.

**Returns**

undefined

FlashEventHandler

The constructor that creates a new FlashEventHandler.

**Syntax**

```
new FlashEventHandler()
```

**Returns**

A FlashEventHandler object.
FlashMovie

Acrobat 9.0

An object that represents a Flash movie in the Scene.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alignMode</td>
<td>integer</td>
<td>R/W</td>
<td>A bit flag that sets the alignment of the movie within the Scene. Values are +1 (left aligned), +2 (right aligned), +4 (top aligned), and +8 (bottom aligned).</td>
</tr>
<tr>
<td>ALIGN_MODE_BOTTOM</td>
<td>string</td>
<td>R</td>
<td>A string constant for the FlashMovie scaleMode type of &quot;bottom&quot;.</td>
</tr>
<tr>
<td>ALIGN_MODE_LEFT</td>
<td>string</td>
<td>R</td>
<td>A string constant for the FlashMovie scaleMode type of &quot;left&quot;.</td>
</tr>
<tr>
<td>ALIGN_MODE_RIGHT</td>
<td>string</td>
<td>R</td>
<td>A string constant for the FlashMovie scaleMode type of &quot;right&quot;.</td>
</tr>
<tr>
<td>ALIGN_MODE_TOP</td>
<td>string</td>
<td>R</td>
<td>A string constant for the FlashMovie scaleMode type of &quot;top&quot;.</td>
</tr>
<tr>
<td>backgroundColor</td>
<td>integer</td>
<td>R/W</td>
<td>Override the background color of a movie. An integer of the form ((\text{red} \times 65536 + \text{green} \times 256 + \text{blue})). Use a value of -1 for the default movie color. The values for \text{red}, \text{green} and \text{blue} are integers between 0 and 255, inclusive, and represent the color components of red, green, and blue, respectively, in the RGB color model.</td>
</tr>
<tr>
<td>desiredResolutionX</td>
<td>integer</td>
<td>R/W</td>
<td>The desired resolution width for the FlashMovie content to be rendered at.</td>
</tr>
<tr>
<td>desiredResolutionY</td>
<td>integer</td>
<td>R/W</td>
<td>The desired resolution height for the FlashMovie content to be rendered at.</td>
</tr>
<tr>
<td>frameNum</td>
<td>integer</td>
<td>R/W</td>
<td>The frame number of the currently displayed frame of the movie. Setting this property advances or rewinds the movie.</td>
</tr>
<tr>
<td>hitEnabled</td>
<td>Boolean</td>
<td>R/W</td>
<td>Determines whether mouse events travel through the FlashMovie to elements in the scene behind it. If true, mouse events are trapped.</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Access</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>---------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>id</td>
<td>integer</td>
<td>R</td>
<td>A unique ID for each FlashMovie in the scene.</td>
</tr>
<tr>
<td>loop</td>
<td>Boolean</td>
<td>R/W</td>
<td>A flag that determines whether the animation loops. If true, the animation loops. If false, it plays only once.</td>
</tr>
<tr>
<td>opacity</td>
<td>number</td>
<td>R/W</td>
<td>The opacity of the FlashMovie represented by a value from 0.0 to 1.0, where 1.0 is completely opaque.</td>
</tr>
<tr>
<td>percentLoaded</td>
<td>integer</td>
<td>R</td>
<td>The percent of the Adobe Flash Player movie that has streamed into the browser so far with possible values from from 0 to 100.</td>
</tr>
<tr>
<td>playing</td>
<td>Boolean</td>
<td>R</td>
<td>A flag that detects whether the movie is currently playing. If true, it is playing. If false, it is paused.</td>
</tr>
<tr>
<td>quality</td>
<td>integer</td>
<td>R/W</td>
<td>The current rendering quality. Permitted values are 0 (Low), 1 (High), 2 (AutoLow), and 3 (AutoHigh).</td>
</tr>
<tr>
<td>readyState</td>
<td>integer</td>
<td>R</td>
<td>The state of the FlashMovie. Permitted values are 0 (Loading), 1 (Uninitialized), 2 (Loaded), 3 (Interactive), 4 (Complete).</td>
</tr>
<tr>
<td>resolutionType</td>
<td>string</td>
<td>R/W</td>
<td>A string value that specifies the type of resolution to be used for the movie. Recognized values are &quot;custom&quot;, &quot;movie&quot;, and &quot;window&quot;.</td>
</tr>
<tr>
<td>RESOLUTION_TYPE_CUSTOM</td>
<td>string</td>
<td>R</td>
<td>A string constant for the FlashMovie resolution type of &quot;custom&quot;.</td>
</tr>
<tr>
<td>RESOLUTION_TYPE.Movie</td>
<td>string</td>
<td>R</td>
<td>A string constant for the FlashMovie resolution type of &quot;movie&quot;.</td>
</tr>
<tr>
<td>RESOLUTION_TYPE_WINDOW</td>
<td>string</td>
<td>R</td>
<td>A string constant for the FlashMovie resolution type of &quot;window&quot;.</td>
</tr>
<tr>
<td>scaleMode</td>
<td>string</td>
<td>R/W</td>
<td>The scale mode of the movie. The value of this property may be &quot;exact fit&quot;, &quot;no border&quot;, or &quot;show all&quot;.</td>
</tr>
<tr>
<td>SCALE_MODE_EXACT_FIT</td>
<td>string</td>
<td>R</td>
<td>A string constant for the FlashMovie scaleMode type of &quot;exact fit&quot;.</td>
</tr>
<tr>
<td>SCALE_MODE_NO_BORDER</td>
<td>string</td>
<td>R</td>
<td>A string constant for the FlashMovie's scaleMode type of &quot;no border&quot;.</td>
</tr>
</tbody>
</table>
FlashMovie

Creates a new FlashMovie from a Resource of type "flash".

Syntax

FlashMovie(FlashMovieResource)

Parameters

FlashMovieResource  A Resource of type "flash".

Returns

A FlashMovie object.

call

Calls into ActionScript with the ExternalInterface calling convention to an exposed method (ExternalInterface.addCallback in ActionScript). The call method returns the return value of the method specified as the first parameter.

Note: The JavaScript for Acrobat API Reference has the callAS method of the AnnotRichMedia object that uses the same mechanism as the call method.

Syntax

call(functionName, [argument1[, ...,argumentn]])
Parameters

<table>
<thead>
<tr>
<th>functionName</th>
<th>A string representing the function name to call in the FlashMovie ActionScript engine.</th>
</tr>
</thead>
<tbody>
<tr>
<td>argument1,argument2,...,argumentn</td>
<td>A comma-delimited list of arguments of varying type to be passed to the function in ActionScript.</td>
</tr>
</tbody>
</table>

Returns

The return value from the called function, which can be of any type.

getVariable

A method that returns the value of the Flash variable specified by varName, and returns undefined if the variable does not exist.

Syntax

getVariable(varName)

Parameters

| varName          | A string representing the variable requested. |

Returns

A string representing the value of the specified Flash variable, or undefined.

gotoFrame

Activates the frame number specified by frameNumber in the current movie. If the data for a requested frame is not yet available, the player goes to the last frame available and stops, causing unexpected results during playback. Use the percentLoaded property to determine if enough of the movie is available to execute the gotoFrame() method. The argument frameNumber is zero-based; that is, frameNumber is 0 in the first frame of the movie, 1 for the second frame, and so on. This differs from the Goto action within Flash, which begins at 1.

Syntax

gotoFrame(frameNumber)

Parameters

| frameNumber      | An integer representing the frame number. |

Returns

undefined
isPlaying

A method that returns true if the movie is currently playing.

Syntax

isPlaying()

Returns

A Boolean type, true if the movie is playing, false otherwise.

pan

This method pans a zoomed-in movie to the coordinates specified by x and y. Use mode to specify whether the values for x and y are pixels or a percentage of the window. The pan method does not pan beyond the boundaries of the zoomed-in movie.

Syntax

pan(x, y, mode)

Parameters

<table>
<thead>
<tr>
<th>x</th>
<th>An integer representing the x coordinate.</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>An integer representing the y coordinate.</td>
</tr>
<tr>
<td>mode</td>
<td>When mode is 0, the coordinates are pixels; when mode is 1, the coordinates are a percentage of the window.</td>
</tr>
</tbody>
</table>

Returns

undefined

play

Starts playing the movie.

Syntax

play()

Returns

undefined

rewind

Goes to the first frame.
Syntax

```
rewind()
```

Returns

```
undefined
```

**setVariable**

Sets the value of the Flash variable specified by `variableName` to the value specified by `value`.

Syntax

```
setVariable(varName, value)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>varName</code></td>
<td>A string representing the variable requested.</td>
</tr>
<tr>
<td><code>value</code></td>
<td>A string value to be set for the provided variable name.</td>
</tr>
</tbody>
</table>

Returns

```
undefined
```

**setZoomRect**

Zooms in on a rectangular area of the movie. The units of the coordinates are measured in twips (1440 units per inch).

**Note:** To calculate the dimensions of a rectangle in the correct units, set the ruler units to Points and multiply the coordinates by 20 to get twips. (There are 72 points per inch.)

Syntax

```
setZoomRect(left, top, right, bottom)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>left</code></td>
<td>An integer representing the left side of the rectangle.</td>
</tr>
<tr>
<td><code>top</code></td>
<td>An integer representing the top side of the rectangle.</td>
</tr>
<tr>
<td><code>right</code></td>
<td>An integer representing the right side of the rectangle.</td>
</tr>
<tr>
<td><code>bottom</code></td>
<td>An integer representing the bottom side of the rectangle.</td>
</tr>
</tbody>
</table>

Returns

```
undefined
```
stop

Stops playing the movie.

Syntax

stop()

Returns

undefined

zoom

This method zooms the view by a relative scale factor specified by percentage. For example, `zoom(50)` doubles the size of the objects in the view, `zoom(200)` reduces the size of objects in the view by one half, and `zoom(0)` resets the view to 100%. You cannot specify a scale factor that will zoom-out the original content further than 100%.

Syntax

zoom(percentage)

Parameters

| percentage | An integer representing the zoom factor. |

Returns

undefined
HitInfo

The object returned when a hit test occurs during a MouseEvent.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>distance</td>
<td>number</td>
<td>R</td>
<td>The distance from the Camera to the HitInfo object’s position.</td>
</tr>
<tr>
<td>material</td>
<td>material</td>
<td>R</td>
<td>Acrobat 8.1</td>
</tr>
<tr>
<td>position</td>
<td>vector3</td>
<td>R</td>
<td>The position of the point where the hit occurred.</td>
</tr>
<tr>
<td>surfaceNormal</td>
<td>vector3</td>
<td>R</td>
<td>Acrobat 8.1</td>
</tr>
<tr>
<td>target</td>
<td>node</td>
<td>R</td>
<td>The target of the hit test.</td>
</tr>
<tr>
<td>textureCoordinate</td>
<td>vector3</td>
<td>R</td>
<td>Acrobat 8.1</td>
</tr>
</tbody>
</table>

Host

Acrobat 7.0.7

An object that provides access to the JavaScript engine context and to pertinent objects within it. The variable host is a global reference to this object. It is a reference to the JavaScript Doc object in which the 3D annotation is contained.
Image

An object that represents an image.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>height</td>
<td>number</td>
<td>R</td>
<td>The image's height, measured in pixels.</td>
</tr>
<tr>
<td>width</td>
<td>number</td>
<td>R</td>
<td>The image's width, measured in pixels.</td>
</tr>
</tbody>
</table>

Image

A constructor that creates a new Image object.

Syntax

```javascript
new Image(resource)
```

Parameters

| resource     | An Image object used to create the new object. |

Returns

An Image object.

See Image for more information about the return object.
**KeyEvent**

An object that is passed as an argument to the `onEvent` method of the `KeyEventHandler` object.

**Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>canvas</td>
<td>canvas</td>
<td>R</td>
<td>The Canvas in which the KeyEvent took place.</td>
</tr>
<tr>
<td>canvasPixelHeight</td>
<td>integer</td>
<td>R</td>
<td>The height, measured in pixels, of the Canvas.</td>
</tr>
<tr>
<td>canvasPixelWidth</td>
<td>integer</td>
<td>R</td>
<td>The width, measured in pixels, of the Canvas.</td>
</tr>
<tr>
<td>characterCode</td>
<td>integer</td>
<td>R</td>
<td>The value of the character pressed according to Acrobat's character mapping, as per this listing of Acrobat character codes:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#</th>
<th>Keys</th>
<th>#</th>
<th>Keys</th>
<th>#</th>
<th>Keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>Left</td>
<td>65</td>
<td>A</td>
<td>97</td>
<td>a</td>
</tr>
<tr>
<td>29</td>
<td>Right</td>
<td>66</td>
<td>B</td>
<td>98</td>
<td>b</td>
</tr>
<tr>
<td>30</td>
<td>Down</td>
<td>67</td>
<td>C</td>
<td>99</td>
<td>c</td>
</tr>
<tr>
<td>31</td>
<td>Up</td>
<td>68</td>
<td>D</td>
<td>100</td>
<td>d</td>
</tr>
<tr>
<td>32</td>
<td>Space</td>
<td>69</td>
<td>E</td>
<td>101</td>
<td>e</td>
</tr>
<tr>
<td>70</td>
<td>F</td>
<td>71</td>
<td>G</td>
<td>103</td>
<td>g</td>
</tr>
<tr>
<td>72</td>
<td>H</td>
<td>73</td>
<td>I</td>
<td>105</td>
<td>i</td>
</tr>
<tr>
<td>74</td>
<td>J</td>
<td>75</td>
<td>K</td>
<td>107</td>
<td>k</td>
</tr>
<tr>
<td>76</td>
<td>L</td>
<td>77</td>
<td>M</td>
<td>109</td>
<td>m</td>
</tr>
<tr>
<td>78</td>
<td>N</td>
<td>79</td>
<td>O</td>
<td>111</td>
<td>o</td>
</tr>
<tr>
<td>80</td>
<td>P</td>
<td>81</td>
<td>Q</td>
<td>113</td>
<td>q</td>
</tr>
<tr>
<td>82</td>
<td>R</td>
<td>83</td>
<td>S</td>
<td>115</td>
<td>s</td>
</tr>
<tr>
<td>84</td>
<td>T</td>
<td>85</td>
<td>U</td>
<td>117</td>
<td>u</td>
</tr>
<tr>
<td>86</td>
<td>V</td>
<td>87</td>
<td>W</td>
<td>119</td>
<td>w</td>
</tr>
<tr>
<td>88</td>
<td>X</td>
<td>89</td>
<td>Y</td>
<td>121</td>
<td>y</td>
</tr>
<tr>
<td>90</td>
<td>Z</td>
<td>91</td>
<td></td>
<td>122</td>
<td>z</td>
</tr>
</tbody>
</table>
### ctrlKeyDown
- **Type**: Boolean
- **Access**: R
- **Description**: Determines whether the Ctrl key (Windows) or Command key (Mac OS) was pressed.

*Note: Acrobat intercepts many of the Ctrl + key events because they are used for accelerators in the main application.*

### currentTool
- **Type**: string
- **Access**: R
- **Description**: The name of the current tool.

### shiftKeyDown
- **Type**: Boolean
- **Access**: R
- **Description**: Determines whether the Shift key was pressed.

*Note: Holding down the shift key changes the value of the KeyEvent.characterCode property.*
**KeyEventHandler**

An object that exposes a callback mechanism that allows a function to be evaluated when a key event occurs. Event handlers are registered with the Runtime `addEventHandler` method.

**Syntax**

```javascript
new KeyEventHandler()
```

**Returns**

A `KeyEventHandler` object.

**onEvent**

A method that is called when a key is pressed.

**Syntax**

```javascript
onEvent(event)
```

**Parameters**

- `event` A `KeyEvent` object representing the event.

**Returns**

undefined
A Node object that illuminates meshes in the Scene.

There are different types of Light objects, each with their own distinct behavior. Infinite Light objects behave much like sunlight in that they cast parallel light in a given direction. Spot Light objects have a fixed cone angle that limits their beam to a conical projection. Point Light objects act similarly to a light bulb, where the light comes from a specific location in 3D space. Currently, none of the Light objects cast shadows.

In addition to the methods and properties that follow, the Light object also contains the same methods and properties as a Node.

### Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>attenuationA</td>
<td>number</td>
<td>R/W</td>
<td>The a coefficient for attenuationType &quot;abc&quot;.</td>
</tr>
<tr>
<td>attenuationB</td>
<td>number</td>
<td>R/W</td>
<td>The b coefficient for attenuationType &quot;abc&quot;.</td>
</tr>
<tr>
<td>attenuationC</td>
<td>number</td>
<td>R/W</td>
<td>The c coefficient for attenuationType &quot;abc&quot;.</td>
</tr>
</tbody>
</table>
| attenuationType   | string    | R/W    | The style of attenuation for the Light object being represented. Attenuation determines how fast the light intensity decreases with distance. The attenuation type of "abc" uses the equation $1 / \max( (a + bd + cdd), 1 )$ to determine the intensity where $d$ is the distance from the light. One of the following values may be assigned: 
  - "abc".
  - "none"
<p>| ATTENUATION_ABC   | string    | R      | Acrobat 7.0.7                                                               |
|                   |           |        | A string constant for the attenuationType of &quot;abc&quot;.                         |
| ATTENUATION_NONE  | string    | R      | Acrobat 7.0.7                                                               |
|                   |           |        | A string constant for the attenuationType of &quot;none&quot;.                        |
| brightness        | number    | R/W    | Specifies the brightness of the emission from the Light. A value of 1 represents a brightness of 100%, though the property may be assigned higher values. |
| color             | Color     | R      | Specifies the color of the light.                                           |</p>
<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>direction</td>
<td>Vector3</td>
<td>R</td>
<td>The direction toward which the light is pointing.</td>
</tr>
<tr>
<td>directionLocal</td>
<td>Vector3</td>
<td>R</td>
<td>Acrobat 7, but not documented until Acrobat 8.1. The direction toward which the light is pointing relative to its parent Node.</td>
</tr>
<tr>
<td>innerConeAngle</td>
<td>number</td>
<td>R/W</td>
<td>The angle, measured in radians, about the direction in which the light is of uniform full density.</td>
</tr>
<tr>
<td>innerRadius</td>
<td>number</td>
<td>R/W</td>
<td>The distance within which the light is of uniform full density.</td>
</tr>
<tr>
<td>outerConeAngle</td>
<td>number</td>
<td>R/W</td>
<td>The angle, measured in radians, about the direction outside of which the light's intensity is zero.</td>
</tr>
<tr>
<td>outerRadius</td>
<td>number</td>
<td>R/W</td>
<td>The distance beyond which the light's intensity is zero.</td>
</tr>
<tr>
<td>position</td>
<td>Vector3</td>
<td>R</td>
<td>The position of the Light object.</td>
</tr>
<tr>
<td>positionLocal</td>
<td>Vector3</td>
<td>R</td>
<td>The position of the Light object relative to its parent Node.</td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td>R/W</td>
<td>The type of Light object being represented. One of the following values may be assigned:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;point&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;spot&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;infinite&quot;</td>
</tr>
<tr>
<td>TYPE_INFINITE</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the Light type of &quot;infinite&quot;.</td>
</tr>
<tr>
<td>TYPE_POINT</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the Light type of &quot;point&quot;.</td>
</tr>
<tr>
<td>TYPE_SPOT</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the Light type of &quot;spot&quot;.</td>
</tr>
</tbody>
</table>
Material

A SceneObject that controls the appearance of materials using the fixed function shader. In addition to the properties below, it contains the same methods and properties as a SceneObject.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ambientColor</td>
<td>Color</td>
<td>R</td>
<td>The ambient color.</td>
</tr>
<tr>
<td>ambientTexture</td>
<td>Texture</td>
<td>R</td>
<td>The ambient texture.</td>
</tr>
<tr>
<td>bumpTexture</td>
<td>Texture</td>
<td>R</td>
<td>A texture map whose value is used to describe the roughness of the object.</td>
</tr>
<tr>
<td>diffuseColor</td>
<td>Color</td>
<td>R</td>
<td>The matte color of an object.</td>
</tr>
<tr>
<td>diffuseTexture</td>
<td>Texture</td>
<td>R</td>
<td>A texture map that is used for the matte color of the object.</td>
</tr>
<tr>
<td>emissiveColor</td>
<td>Color</td>
<td>R</td>
<td>Emissive color except that it does not require that any lighting to display. An object with an emissive color of white and no texture will appear pure white in the scene.</td>
</tr>
<tr>
<td>emissiveTexture</td>
<td>Texture</td>
<td>R</td>
<td>The emissive texture.</td>
</tr>
<tr>
<td>opacity</td>
<td>number</td>
<td>R/W</td>
<td>The total opacity of the material.</td>
</tr>
<tr>
<td>opacityTexture</td>
<td>Texture</td>
<td>R</td>
<td>A texture map whose brightness is used for the level of opacity of the object. White signifies completely opaque while black signifies completely transparent.</td>
</tr>
<tr>
<td>phongExponent</td>
<td>number</td>
<td>R/W</td>
<td>The Phong exponent. The Phong exponent defines the “tightness” of the highlight. A higher exponent results in a smaller, tighter highlight while a lower exponent results in a broader flatter one.</td>
</tr>
<tr>
<td>reflectionStrength</td>
<td>number</td>
<td>R/W</td>
<td>The reflection level, which can be a value from 0.0 to 1.0.</td>
</tr>
<tr>
<td>reflectionTexture</td>
<td>Texture</td>
<td>R</td>
<td>The reflection texture, also known as an environment map, the texture is used to store the image of the environment surrounding the rendered object. It simulates the reflection of a mirrored surface.</td>
</tr>
</tbody>
</table>
**attachFlashMovie**

**Acrobat 9.0**

Sets the provided FlashMovie as the diffuse texture for the Material.

**Syntax**

```javascript
attachFlashMovie(movie)
```

**Parameters**

- `movie` The FlashMovie object to be used as the diffuse texture.

**Returns**

undefined

---

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>specularColor</td>
<td>Color</td>
<td>R</td>
<td>The specular color. The specularColor is the color of the highlight on the material.</td>
</tr>
<tr>
<td>specularStrength</td>
<td>number</td>
<td>R/W</td>
<td>The specular strength, which is a measure of how shiny the material is.</td>
</tr>
</tbody>
</table>
Matrix4x4

A four-by-four matrix commonly used for transformations.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>determinant</td>
<td>number</td>
<td>R/W</td>
<td>The determinant of the matrix.</td>
</tr>
<tr>
<td>inverse</td>
<td>Matrix4x4</td>
<td>R</td>
<td>The inverse of the matrix.</td>
</tr>
<tr>
<td>scaleComponent</td>
<td>Vector3</td>
<td>R</td>
<td>The scale component of the transformation.</td>
</tr>
<tr>
<td>translation</td>
<td>Vector3</td>
<td>R</td>
<td>The translation component of the transformation.</td>
</tr>
<tr>
<td>transpose</td>
<td>Matrix4x4</td>
<td>R</td>
<td>The transpose of the matrix.</td>
</tr>
</tbody>
</table>

Matrix4x4

A constructor that creates a new Matrix4x4 object initialized to the identity matrix.

Syntax

```javascript
new Matrix4x4()
```

Returns

A Matrix4x4 object initialized to the identity matrix.

Matrix4x4

A constructor that creates a new Matrix4x4 object initialized to the specified matrix.

Syntax

```javascript
new Matrix4x4(matrix)
```

Parameters

| matrix            | A Matrix4x4 object used to initialize the new matrix. |

Returns

A Matrix4x4 object initialized to the specified matrix.
invertInPlace

Inverts the matrix.

Returns

undefined

isEqual

Determines whether the current matrix is equal to the specified matrix.

Syntax

isEqual(matrix)

Parameters

matrix  A Matrix4x4 object used for the comparison.

Returns

Returns true if the matrices are equal, false otherwise.

multiply

Multiplies the current matrix by the specified matrix.

Syntax

multiply(matrix)

Parameters

matrix  A Matrix4x4 object used for the multiplication.

Returns

A Matrix4x4 object.

multiplyInPlace

Multiplies the current matrix by the specified matrix, and updates the current matrix with the resulting value.

Syntax

multiplyInPlace(matrix)
**Parameters**

matrix A Matrix4x4 object used for the multiplication.

**Returns**

undefined

**rotateWithQuaternion**

Rotates the current matrix using the specified Quaternion.

**Syntax**

`rotateWithQuaternion(quaternion)`

**Parameters**

quaternion A Quaternion object used for the rotation.

**Returns**

A Matrix4x4 object.

**rotateWithQuaternionInPlace**

Rotates the current matrix using the specified quaternion, and updates the current matrix with the resulting value.

**Syntax**

`rotateWithQuaternionInPlace(quaternion)`

**Parameters**

quaternion A Quaternion object used for the rotation.

**Returns**

undefined

**rotateAboutLine**

Rotates the current matrix about the specified line.

**Syntax**

`rotateAboutLine(angle, start, end)`
Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>angle</td>
<td>The angle of rotation, in radians.</td>
</tr>
<tr>
<td>start</td>
<td>A point described by a Vector3 object used to specify the beginning of the line of rotation, which is represented by start - end.</td>
</tr>
<tr>
<td>end</td>
<td>A point described by a Vector3 object used to specify the end of the line of rotation, which is represented by start - end.</td>
</tr>
</tbody>
</table>

Returns

A Matrix4x4 object.

rotateAboutLineInPlace

Rotates the current matrix about the specified line, and updates the current matrix with the resulting value.

Syntax

rotateAboutLineInPlace(angle, start, end)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>angle</td>
<td>The angle of rotation, in radians.</td>
</tr>
<tr>
<td>start</td>
<td>A Vector3 object used to specify the line of rotation, which is represented by start - end.</td>
</tr>
<tr>
<td>end</td>
<td>A Vector3 object used to specify the line of rotation, which is represented by start - end.</td>
</tr>
</tbody>
</table>

Returns

undefined

rotateAboutX

Rotates the current matrix about the x axis.

Syntax

rotateAboutX(angle)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>angle</td>
<td>The angle of rotation, in radians.</td>
</tr>
</tbody>
</table>
Returns

A Matrix4x4 object.

rotateAboutXInPlace

Rotates the current matrix about the x axis, and updates the current matrix with the resulting value.

Syntax

rotateAboutXInPlace(angle)

Parameters

angle The angle of rotation, in radians.

Returns

undefined

rotateAboutVector

Rotates the current matrix about the specified vector.

Syntax

rotateAboutVector(angle, axis)

Parameters

angle The angle of rotation, in radians.
axis A Vector3 object about which the matrix is rotated.

Returns

A Matrix4x4 object.

rotateAboutVectorInPlace

Rotates the current matrix about the specified vector, and updates the current matrix with the resulting value.

Syntax

rotateAboutVectorInPlace(angle, axis)
Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>angle</td>
<td>The angle of rotation, in radians.</td>
</tr>
<tr>
<td>axis</td>
<td>A Vector3 object about which the matrix is rotated.</td>
</tr>
</tbody>
</table>

Returns

undefined

rotateAboutY

Rotates the current matrix about the y axis.

Syntax

rotateAboutY(angle)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>angle</td>
<td>The angle of rotation, in radians.</td>
</tr>
</tbody>
</table>

Returns

A Matrix4x4 object.

rotateAboutYInPlace

Rotates the current matrix about the y axis, and updates the current matrix with the resulting value.

Syntax

rotateAboutYInPlace(angle)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>angle</td>
<td>The angle of rotation, in radians.</td>
</tr>
</tbody>
</table>

Returns

undefined

rotateAboutZ

Rotates the current matrix about the z axis.
Syntax
rotateAboutZ(angle)

Parameters

angle The angle of rotation, in radians.

Returns
A Matrix4x4 object.

rotateAboutZInPlace

Rotates the current matrix about the z axis, and updates the current matrix with the resulting value.

Syntax
rotateAboutZInPlace(angle)

Parameters

angle The angle of rotation, in radians.

Returns
undefined

cscale

Scales the current matrix using the specified scaling components.

Syntax
cscale(x, y, z)

Parameters

x The scaling component in the x direction.
y The scaling component in the y direction.
z The scaling component in the z direction.

Returns
A Matrix4x4 object.
scaleInPlace

Scales the current matrix using the specified scaling components, and updates the current matrix with the resulting value.

Syntax

scaleInPlace(x, y, z)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>The scaling component in the x direction.</td>
</tr>
<tr>
<td>y</td>
<td>The scaling component in the y direction.</td>
</tr>
<tr>
<td>z</td>
<td>The scaling component in the z direction.</td>
</tr>
</tbody>
</table>

Returns

undefined

set

Sets the value of the current matrix using the specified matrix.

Syntax

set(matrix)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>matrix</td>
<td>The matrix whose value is copied into the current matrix.</td>
</tr>
</tbody>
</table>

Returns

undefined

set

Acrobat 8.1

Sets the value of the current matrix using an array.

Syntax

set( array )
Parameters

| array | The array of length 16 whose values are copied into the current matrix. |

Returns

undefined

set

Acrobat 8.1

Sets the value of the current matrix using 16 numeric values.

Syntax


Parameters

v0-v15 | Number values for the given indices of the matrix. |

Returns

undefined

setIdentity

Sets the value of the current matrix to the identity matrix.

Syntax

setIdentity()

Returns

undefined

setView

Sets the current matrix according to the specified component vectors.

Syntax

setView(position, direction, up)
**Parameters**

- **position**
  A `Vector3` object used to specify the position component.

- **direction**
  A `Vector3` object used to specify the direction component.

- **up**
  A `Vector3` object used to specify the upward component.

**Returns**

(undefined)

### transformDirection

Transforms the specified vector by the current matrix.

**Syntax**

```
transformDirection(vector)
```

**Parameters**

- **vector**
  The `Vector3` object to be transformed.

**Returns**

A `Vector3` object.

### transformPosition

Transforms the specified position by the current matrix.

**Syntax**

```
transformPosition(position)
```

**Parameters**

- **position**
  A `Vector3` object representing the position to be transformed.

**Returns**

A `Vector3` object.

### translate

Translates the current matrix by the components of the specified vector.
Syntax
```
translate(translation)
```

Parameters

- `translation` The `Vector3` object whose components are used to perform the matrix translation.

Returns

- A `Matrix4x4` object.

**translateInPlace**

Translates the current matrix by the components of the specified vector, and updates the current matrix with the resulting value.

Syntax
```
translateInPlace(translation)
```

Parameters

- `translation` The `Vector3` object whose components are used to perform the matrix translation.

Returns

- undefined

**transposeInPlace**

Sets the value of the current matrix to its transpose.

Syntax
```
transposeInPlace()
```

Returns

- undefined
MenuEvent

An object that is passed as an argument to the `onEvent` method of the `MenuEventHandler` object.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>canvas</td>
<td>Canvas</td>
<td>R</td>
<td>The Canvas in which the <code>MenuEvent</code> took place.</td>
</tr>
<tr>
<td>currentTool</td>
<td>string</td>
<td>R</td>
<td>The name of the current tool.</td>
</tr>
<tr>
<td>menuItemChecked</td>
<td>Boolean</td>
<td>R</td>
<td>Determines whether the menu item was selected.</td>
</tr>
<tr>
<td>menuItemName</td>
<td>string</td>
<td>R</td>
<td>The name of the selected menu item.</td>
</tr>
</tbody>
</table>
MenuEventHandler

A MenuEventHandler object exposes a callback mechanism that allows a function to be evaluated when an event occurs. Event handlers are registered with the Runtime addEventHandler method.

MenuEventHandler

A constructor that creates a new MenuEventHandler object.

Syntax

```javascript
new MenuEventHandler()
```

Returns

A MenuEventHandler object.

onEvent

A method that is called when a custom menu item is selected on the context menu for an active 3D annotation.

Syntax

```javascript
onEvent(event)
```

Parameters

- event: A MenuEvent object representing the event.

Returns

undefined
Mesh

A Node object that contains geometry. A Mesh object with no geometry has children Node objects that can be transformed as a group. In addition to the methods and properties below, it contains the same methods and properties as a Node.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>faceCount</td>
<td>number object</td>
<td>The number of faces the 3D mesh has.</td>
</tr>
<tr>
<td>material</td>
<td>material</td>
<td>The Mesh object's default Material.</td>
</tr>
<tr>
<td>renderMode</td>
<td>string</td>
<td>The Mesh object's rendering style, which can be one of the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● &quot;default&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● &quot;bounding box&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● &quot;transparent bounding box&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● &quot;transparent bounding box outline&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● &quot;vertices&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● &quot;shaded vertices&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● &quot;wireframe&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● &quot;shaded wireframe&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● &quot;solid&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● &quot;transparent&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● &quot;solid wireframe&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● &quot;transparent wireframe&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● &quot;illustration&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● &quot;solid outline&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● &quot;shaded illustration&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● &quot;hidden wireframe&quot;</td>
</tr>
</tbody>
</table>

computeBoundingBox

Acrobat 7.0.7

Computes the bounds of the Node object.

Syntax

computeBoundingBox()  

Returns

A BoundingBox object.
setColor

Acrobat 11

Sets the color, either for the entire mesh or for any one of the faces of the mesh. `setColor` can be called several times for the same mesh, either to change the color of the entire mesh or to change the color of the faces.

Syntax

setColor(color, faceIndex)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>color</td>
<td>(Optional) A Color object representing the desired color. Omit this parameter to reset the color and return to the original color (the one read from the PRC or U3D file).</td>
</tr>
<tr>
<td>faceIndex</td>
<td>(Optional) The index representing the face whose color is to be changed. Omit this parameter to change the color of the entire mesh. If <code>faceIndex</code> is out of bounds, no action is performed by this method.</td>
</tr>
</tbody>
</table>

Returns

undefined.
MouseEvent

An object that is passed as an argument to the `onEvent` method of the `MouseEventHandler` object.

### Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>canvas</td>
<td>Canvas</td>
<td>R</td>
<td>The Canvas in which the MouseEvent took place.</td>
</tr>
<tr>
<td>canvasPixelHeight</td>
<td>integer</td>
<td>R</td>
<td>The height, measured in pixels, of the Canvas in which the MouseEvent took place.</td>
</tr>
<tr>
<td>canvasPixelWidth</td>
<td>integer</td>
<td>R</td>
<td>The width, measured in pixels, of the Canvas in which the MouseEvent took place.</td>
</tr>
<tr>
<td>ctrlKeyDown</td>
<td>Boolean</td>
<td>R</td>
<td>Determines whether the Ctrl key (Windows) or Command key (Mac OS) was pressed.</td>
</tr>
<tr>
<td>currentTool</td>
<td>string</td>
<td>R</td>
<td>The name of the current tool.</td>
</tr>
<tr>
<td>hits</td>
<td>Array</td>
<td>R</td>
<td>A set of HitInfo objects ordered by distance from nearest to furthest.</td>
</tr>
<tr>
<td>isDoubleClick</td>
<td>Boolean</td>
<td>R</td>
<td>Determines whether a double-click event occurred.</td>
</tr>
<tr>
<td>isMouseDown</td>
<td>Boolean</td>
<td>R</td>
<td>Determines whether the mouse button was pressed.</td>
</tr>
<tr>
<td>isMouseHit</td>
<td>Boolean</td>
<td>R</td>
<td>Determines whether the target is under the mouse cursor.</td>
</tr>
<tr>
<td>isMouseMove</td>
<td>Boolean</td>
<td>R</td>
<td>Determines whether the mouse position changed.</td>
</tr>
<tr>
<td>isMouseOut</td>
<td>Boolean</td>
<td>R</td>
<td>Determines whether the mouse position moved off the target.</td>
</tr>
<tr>
<td>isMouseOver</td>
<td>Boolean</td>
<td>R</td>
<td>Determines whether the mouse position moved onto the target.</td>
</tr>
<tr>
<td>isMouseUp</td>
<td>Boolean</td>
<td>R</td>
<td>Determines whether the mouse button was released.</td>
</tr>
<tr>
<td>leftButtonDown</td>
<td>Boolean</td>
<td>R</td>
<td>Determines whether the left mouse button was pressed.</td>
</tr>
<tr>
<td>mouseX</td>
<td>integer</td>
<td>R</td>
<td>The x position of the mouse cursor in the Canvas.</td>
</tr>
<tr>
<td>mouseY</td>
<td>integer</td>
<td>R</td>
<td>The y position of the mouse cursor in the Canvas.</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Access</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| rightButtonDown | Boolean | R      | Version 7.0.1  
Determines whether the right mouse button was pressed.                  |
| shiftKeyDown    | Boolean | R      | Determines whether the Shift key was pressed.                              |
MouseEventHandler

An object that exposes a callback mechanism that allows a function to be evaluated when a mouse event occurs. The handler may be customized to filter out certain event types. Event handlers are registered with the Runtime `addEventListener` method.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>onMouseDoubleClick</td>
<td>Boolean</td>
<td>R/W</td>
<td>When set to true, the handler is called back when a mouse button is clicked twice in rapid succession on the target object. If no target is specified, the handler calls back on any double-click.</td>
</tr>
<tr>
<td>onMouseDown</td>
<td>Boolean</td>
<td>R/W</td>
<td>When set to true, the handler is called back when a mouse button is initially pressed while the cursor is over the target object. If no target is specified, the handler calls back on any button press.</td>
</tr>
<tr>
<td>onMouseHit</td>
<td>Boolean</td>
<td>R/W</td>
<td>When set to true, the handler is called back continuously when the cursor is over the target object. In the case of onMouseHit, it does not matter if the target object is behind another object in the scene. The list of resultant hit objects are provided in the MouseEvent hits property.</td>
</tr>
<tr>
<td>onMouseMove</td>
<td>Boolean</td>
<td>R/W</td>
<td>When set to true, the handler is called back when the cursor moves over the target object. If no target is specified, the handler calls back on any mouse movement over the 3D annotation.</td>
</tr>
<tr>
<td>onMouseOut</td>
<td>Boolean</td>
<td>R/W</td>
<td>When set to true, the handler is called back once when the cursor moves off of the target object. To be called back, the target must be the frontmost object. To exclude objects, use the Node hitEnabled property.</td>
</tr>
<tr>
<td>onMouseOver</td>
<td>Boolean</td>
<td>R/W</td>
<td>When set to true, the handler is called back once when the cursor moves over the target object.</td>
</tr>
<tr>
<td>onMouseUp</td>
<td>Boolean</td>
<td>R/W</td>
<td>When set to true, the handler is called back when a mouse button is initially released. If a target is specified, it calls back only when the cursor is over the handler’s target.</td>
</tr>
</tbody>
</table>
MouseEventHandler

A constructor that creates a new MouseEventHandler object.

Syntax

```javascript
new MouseEventHandler()
```

Returns

A MouseEventHandler object.

onEvent

A method that is called when a mouse event occurs.

Syntax

```javascript
onEvent(event)
```

Parameters

- `event` A MouseEvent object representing the event.

Returns

undefined
Node

An object within the Scene hierarchy (a SceneObject) that has a 3D representation. The following objects are considered Node objects:

- Bone
- Camera
- ClippingPlane
- Dummy
- Light
- Mesh
- Procedural

To obtain a Node object’s type, use the standard JavaScript constructor property. For example, the following syntax prints the Node object’s type to the console:

```javascript
console.println(myNode.constructor.name);
```

In addition to the methods and properties below, it contains the same methods and properties as a SceneObject.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>firstChild</td>
<td>Node (if the first child exists), None otherwise</td>
<td>R</td>
<td>The Node object’s first child.</td>
</tr>
<tr>
<td>hitEnabled</td>
<td>Boolean</td>
<td>R/W</td>
<td>Determines whether the Node is included in hit tests. The default value is true.</td>
</tr>
</tbody>
</table>
| info           | string                      | R      | Acrobat 7.0.7
|                |                              |        | Information associated with the Node.                                      |
| metadataString | string                      | R      | Acrobat 8.1
|                |                              |        | A string containing Node-specific metadata.                                |
| nextSibling    | Node (if the next sibling exists), None otherwise | R      | The next sibling.                                                          |
| opacity        | number                      | R/W    | Acrobat 7.0.7
|                |                              |        | The Node opacity. A value from 0 to 1, where 1 is completely opaque.        |
| parent         | object                      | R      | The Node object’s parent Node or Scene.                                    |
| transform      | Matrix4x4                    | R      | The local to world transformation matrix for the Node.                     |
detachFromCurrentAnimation

Removes the ability of the currently active Animation of the Node object to transform the Node.

Syntax

detachFromCurrentAnimation()

Returns

undefined

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wireframeColor</td>
<td>Color</td>
<td>R</td>
<td>The Color object used to determine the wireframe appearance.</td>
</tr>
<tr>
<td>visible</td>
<td>Boolean</td>
<td>R/W</td>
<td>Determines whether the Node object should be shown. This property applies to mesh notes only. For example, modifying the empty parent node of a mesh tree has no effect on the child mesh tree items. In such cases it is recommended that you modify a parent node that is also a mesh node, and the child mesh items will have the same value for this property.</td>
</tr>
</tbody>
</table>
Procedural

Deprecated

A `Node` object used to represent procedurally created geometry, such as constructive solid geometry (CSG) solids, procedural spheres, or NURB objects (a 3D curve or surface). A `Procedural` object contains the same methods and properties as a `Node`. 
Quaternion

Represents a rotation in 3D space, and allows for smooth interpolation (blending) between orientations of subjects. A Quaternion is typically used for animating a Camera or Mesh over time, and can be converted to and from angles of rotation about the axes.

Syntax

\[
\text{new Quaternion()} \\
\]

Returns

A Quaternion object.

Quaternion

A constructor that initializes the object with the identity matrix.

Syntax

\[
\text{new Quaternion(matrix)} \\
\]

Parameters

- **matrix**: A Matrix4x4 object representing the rotation matrix.

Returns

A Quaternion object.

Quaternion

A constructor that initializes the object with the specified rotation matrix.

Syntax

\[
\text{new Quaternion(quaternion)} \\
\]

Parameters

- **quaternion**: A Quaternion object used to initialize the new object.
Returns
A Quaternion object.

**interpolate**

Creates a Quaternion object interpolated from the current and specified Quaternion objects and a.

**Syntax**

interpolate(quaternion, a)

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>quaternion</td>
<td>A Quaternion object used to interpolate the new object.</td>
</tr>
<tr>
<td>a</td>
<td>A number value, from 0.0 to 1.0, that specifies the degree (percentage) of interpolation. A value of 0.5 represents an interpolation halfway between the current and specified Quaternion objects.</td>
</tr>
</tbody>
</table>

**Returns**
A Quaternion object.

**interpolateInPlace**

Creates a Quaternion object interpolated from the current and specified Quaternion objects and a, and updates the current Quaternion object with the new value.

**Syntax**

interpolateInPlace(quaternion, a)

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>quaternion</td>
<td>A Quaternion object used to interpolate the new object.</td>
</tr>
<tr>
<td>a</td>
<td>A number value, from 0.0 to 1.0, that specifies the degree (percentage) of interpolation. A value of 0.5 represents an interpolation halfway between the current and specified Quaternion objects.</td>
</tr>
</tbody>
</table>

**Returns**
A Quaternion object.

**normalize**

Normalizes the Quaternion object
Syntax
   normalize()

Returns
   undefined
RenderEvent

An object that is passed as an argument to the `onEvent` method of the `RenderEventHandler` object.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>canvas</td>
<td>Canvas</td>
<td>R</td>
<td>The Canvas that is the target of the RenderEvent.</td>
</tr>
<tr>
<td>canvasPixelHeight</td>
<td>integer</td>
<td>R</td>
<td>The height, measured in pixels, of the Canvas for which the RenderEvent is intended.</td>
</tr>
<tr>
<td>canvasPixelWidth</td>
<td>integer</td>
<td>R</td>
<td>The width, measured in pixels, of the Canvas for which the RenderEvent is intended.</td>
</tr>
<tr>
<td>currentTool</td>
<td>string</td>
<td>R</td>
<td>The name of the current tool.</td>
</tr>
</tbody>
</table>
RenderEventHandler

An object that exposes a callback mechanism that allows a function to be evaluated when an event occurs. Event handlers are registered with the Runtime `addEventHandler` method. It issues a callback just before each Canvas is rendered.

**RenderEventHandler**

A constructor that creates a new `RenderEventHandler` object.

**Syntax**

`new RenderEventHandler()`

**Returns**

A `RenderEventHandler` object.

**onEvent**

A method that is called immediately before the Canvas is rendered.

**Syntax**

`onEvent(event)`

**Parameters**

| `event` | A `RenderEvent` object representing the event. |

**Returns**

undefined
## RenderOptions

An object that describes the style with which to render Node objects in the Scene.

### Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boundingBoxColor</td>
<td>Color</td>
<td>R</td>
<td>A Color object to be applied to the bounding box.</td>
</tr>
<tr>
<td>clippingPlaneColor</td>
<td>Color</td>
<td>R</td>
<td>A Color object to be applied to the clipping plane.</td>
</tr>
<tr>
<td>clippingPlaneIntersectionColor</td>
<td>Color</td>
<td>R</td>
<td>A Color object to be applied to the clipping plane intersection.</td>
</tr>
<tr>
<td>defaultAmbientColor</td>
<td>Color</td>
<td>R</td>
<td>A Color object to be applied to the default ambient Material.</td>
</tr>
<tr>
<td>defaultDiffuseColor</td>
<td>Color</td>
<td>R</td>
<td>A Color object to be applied to the default diffuse Material.</td>
</tr>
<tr>
<td>defaultEmissiveColor</td>
<td>Color</td>
<td>R</td>
<td>A Color object to be applied to the default emissive Material.</td>
</tr>
<tr>
<td>defaultSpecularColor</td>
<td>Color</td>
<td>R</td>
<td>A Color object to be applied to the default specular Material.</td>
</tr>
<tr>
<td>illustrationRenderModeFaceColor</td>
<td>Color</td>
<td>R</td>
<td>Acrobat 7.0.7 The color of the faces when the render mode is Illustration.</td>
</tr>
<tr>
<td>illustrationRenderModeLineColor</td>
<td>Color</td>
<td>R</td>
<td>A Color object to be applied to the illustration lines.</td>
</tr>
<tr>
<td>pointsRenderModeColor</td>
<td>Color</td>
<td>R</td>
<td>A Color object to be applied to the vertices in point render mode.</td>
</tr>
<tr>
<td>shadedIllustrationRenderModeLineColor</td>
<td>Color</td>
<td>R</td>
<td>A Color object to be applied to the shaded illustration lines.</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Access</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>solidGridColorEven</td>
<td>Color</td>
<td>R</td>
<td>Acrobat 7.0.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The color of the even squares of the checkered grid when drawn in solid mode.</td>
</tr>
<tr>
<td>solidGridColorOdd</td>
<td>Color</td>
<td>R</td>
<td>Acrobat 7.0.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The color of the odd squares of the checkered grid when drawn in solid mode.</td>
</tr>
<tr>
<td>solidRenderModeLineColor</td>
<td>Color</td>
<td>R</td>
<td>Acrobat 7.0.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>A Color object to be applied to the solid or transparent lines in render mode.</td>
</tr>
<tr>
<td>transparentBoundsRenderModeColor</td>
<td>Color</td>
<td>R</td>
<td>Acrobat 7.0.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>A Color object to be applied to the transparent bounding box.</td>
</tr>
<tr>
<td>transparentGridColorEven</td>
<td>Color</td>
<td>R</td>
<td>Acrobat 7.0.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The color of the even squares of the checkered grid when drawn in transparent mode.</td>
</tr>
<tr>
<td>transparentGridColorOdd</td>
<td>Color</td>
<td>R</td>
<td>Acrobat 7.0.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The color of the odd squares of the checkered grid when drawn in transparent mode.</td>
</tr>
<tr>
<td>wireframeRenderModeColor</td>
<td>Color</td>
<td>R</td>
<td>Acrobat 7.0.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The color of the wires when the render mode is Wireframe.</td>
</tr>
<tr>
<td>xAxisColor</td>
<td>Color</td>
<td>R</td>
<td>Acrobat 7.0.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The color of the x axis.</td>
</tr>
<tr>
<td>yAxisColor</td>
<td>Color</td>
<td>R</td>
<td>Acrobat 7.0.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The color of the y axis.</td>
</tr>
<tr>
<td>zAxisColor</td>
<td>Color</td>
<td>R</td>
<td>Acrobat 7.0.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The color of the z axis.</td>
</tr>
</tbody>
</table>
Resource

An object that creates an abstraction for loading behavior in files and streams.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>string</td>
<td>R</td>
<td>The type of Resource object, which can be one of the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;image&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;model&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;flash&quot; (Acrobat 9.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;unknown&quot;</td>
</tr>
</tbody>
</table>

| TYPE_IMAGE   | string | R      | Acrobat 7.0.7                                                               |
|              |        |        | A string constant for the Resource type of "image".                        |

| TYPE_MODEL   | string | R      | Acrobat 7.0.7                                                               |
|              |        |        | A string constant for the Resource type of "model".                        |

| TYPE_UNKNOWN | string | R      | Acrobat 7.0.7                                                               |
|              |        |        | A string constant for the Resource type of "unknown".                      |

| TYPE_FLASH   | string | R      | Acrobat 9.0                                                                |
|              |        |        | A string constant for the Resource type of "flash".                        |

Resource

A constructor that creates a new Resource.

Syntax

new Resource(pathname)

Parameters

<table>
<thead>
<tr>
<th>parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pathname</td>
<td>A string representing the path of the file or stream. Can load embedded resources only from within the PDF file. The pathname string must start with pdf://.</td>
</tr>
</tbody>
</table>

Returns

A Resource object.
## Runtime

An object that represents the run-time instance of the player. Each `Runtime` object can have its own unique script engine and set of annotations. The variable `runtime` is a global reference to this object.

### Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUTTON_TYPE_PUSH</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the custom tool button type of push button. It is used with the <code>addCustomToolButton</code> method.</td>
</tr>
<tr>
<td>BUTTON_TYPE_TOOL</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the custom button type of tool button. It is used with the <code>addCustomToolButton</code> method.</td>
</tr>
<tr>
<td>canvasCount</td>
<td>number</td>
<td>R</td>
<td>Acrobat 8.1 The number of Canvases that are attached to the active 3D annotation.</td>
</tr>
<tr>
<td>ctrlKeyDown</td>
<td>Boolean</td>
<td>R</td>
<td>Determines whether the Ctrl key (Windows) or Command key (Mac OS) was pressed.</td>
</tr>
<tr>
<td>eventHandlerCount</td>
<td>integer</td>
<td>R</td>
<td>The number of registered event handlers.</td>
</tr>
<tr>
<td>instances</td>
<td>Array</td>
<td>R</td>
<td>Acrobat 7.0.7 An array of JavaScript <code>Annot3D</code> objects that are attached to the 3D script context.</td>
</tr>
<tr>
<td>MENU_ITEM_TYPE_CHECKED</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the custom menu item type of checked. It is used with the <code>addCustomMenuItem</code> method.</td>
</tr>
<tr>
<td>MENU_ITEM_TYPE_DEFAULT</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the custom menu item type of default. It is used with the <code>addCustomMenuItem</code> method.</td>
</tr>
<tr>
<td>overrideFlyTool</td>
<td>Boolean</td>
<td>R/W</td>
<td>Acrobat 9.0 Determines whether to override the built-in Fly tool behavior.</td>
</tr>
<tr>
<td>overrideNavTools</td>
<td>Boolean</td>
<td>R/W</td>
<td>Determines whether to disable all default navigation behavior.</td>
</tr>
</tbody>
</table>

**Note:** Setting this property does not prevent view changes.
<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>overridePanTool</td>
<td>Boolean</td>
<td>R/W</td>
<td>Determines whether to override the built-in Pan tool behavior.</td>
</tr>
<tr>
<td>overrideRotateTool</td>
<td>Boolean</td>
<td>R/W</td>
<td>Determines whether to override the built-in Rotate tool behavior.</td>
</tr>
<tr>
<td>overrideSelection</td>
<td>Boolean</td>
<td>R/W</td>
<td>Determines whether to override the built-in Selection tool behavior.</td>
</tr>
<tr>
<td>overrideSpinTool</td>
<td>Boolean</td>
<td>R/W</td>
<td>Determines whether to override the built-in Spin tool behavior.</td>
</tr>
<tr>
<td>overrideViewChange</td>
<td>Boolean</td>
<td>R/W</td>
<td>Determines whether to override the setting of Views from Acrobat.</td>
</tr>
<tr>
<td>overrideWalkTool</td>
<td>Boolean</td>
<td>R/W</td>
<td>Determines whether to override the built-in Walk tool behavior.</td>
</tr>
<tr>
<td>overrideScrollWheel</td>
<td>Boolean</td>
<td>R/W</td>
<td>Determines whether to override the built-in scroll-wheel behavior.</td>
</tr>
<tr>
<td>overrideZoomTool</td>
<td>Boolean</td>
<td>R/W</td>
<td>Determines whether to override the built-in Zoom tool behavior.</td>
</tr>
<tr>
<td>scrollWheelSpeed</td>
<td>number</td>
<td>R/W</td>
<td>A speed multiplier for the value of the scroll-wheel motion.</td>
</tr>
<tr>
<td>shiftKeyDown</td>
<td>Boolean</td>
<td>R</td>
<td>Determines whether the Shift key was pressed.</td>
</tr>
</tbody>
</table>

**Note:** Setting this property does not affect the pan behavior of other navigation tools.
<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>speedThreshold</td>
<td>number</td>
<td>R/W</td>
<td>Acrobat 8.1&lt;br&gt;A length (based upon the diagonal of the scene's bounding box) under which the Walk tool's motion is scaled relative to the size of the model. The Walk tool's motion is constant based upon the scene's scale factor, such that it emulates a natural pace relative to the model's size. This works well for architectural models that are created with a defined scale. However, the walk motion is too quick for very small models.</td>
</tr>
<tr>
<td>strafeSpeed</td>
<td>number</td>
<td>R/W</td>
<td>Acrobat 8.1&lt;br&gt;A speed multiplier for the lateral motion while using the Walk tool.</td>
</tr>
<tr>
<td>TOOL_NAME_FLY</td>
<td>string</td>
<td>R</td>
<td>Acrobat 9.0&lt;br&gt;A string constant for the name of the fly tool. Its value is &quot;Fly&quot;.</td>
</tr>
<tr>
<td>TOOL_NAME_MEASURE</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7&lt;br&gt;A string constant for the name of the measure tool. Its value is &quot;Measure&quot;.</td>
</tr>
<tr>
<td>TOOL_NAME_PAN</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7&lt;br&gt;A string constant for the name of the pan tool. Its value is &quot;Pan&quot;.</td>
</tr>
<tr>
<td>TOOL_NAME_ROTATE</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7&lt;br&gt;A string constant for the name of the rotate tool. Its value is &quot;Rotate&quot;.</td>
</tr>
<tr>
<td>TOOL_NAME_SPIN</td>
<td>string</td>
<td>R</td>
<td>Acrobat 8.0&lt;br&gt;A string constant for the name of the Spin tool. Its value is &quot;Spin&quot;.</td>
</tr>
<tr>
<td>TOOL_NAME_WALK</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7&lt;br&gt;A string constant for the name of the walk tool. Its value is &quot;Walk&quot;.</td>
</tr>
<tr>
<td>TOOL_NAME_ZOOM</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7&lt;br&gt;A string constant for the name of the zoom tool. Its value is &quot;Zoom&quot;.</td>
</tr>
<tr>
<td>version</td>
<td>number</td>
<td>R</td>
<td>The number corresponding to the version of the Runtime system.</td>
</tr>
</tbody>
</table>
addCustomMenuItem

Creates a custom menu item in the 3D annotation context menu.

Syntax

addCustomMenuItem(name, label, type, checkedState)

Parameters

- name: A string identifying the menu item.
- label: A string appearing on the menu item.
- type: A string indicating whether it is a checked menu item. A checked menu item has a check mark toggle next to it. Its possible values are:
  - "default"
  - "checked"
- checkedState: A Boolean value indicating the state of a checked menu item.

Returns

undefined

addCustomToolButton

Creates a custom tool button in the 3D toolbar.

Syntax

addCustomToolButton(name, label, type)

Parameters

- name: A string identifying the tool button.
Returns

undefined

addEventHandler

Registers the provided event handler.

Syntax

addEventHandler(eventHandler)

Parameters

<table>
<thead>
<tr>
<th>eventHandler</th>
<th>The event handler object to be registered.</th>
</tr>
</thead>
</table>

Returns

undefined

disableTool

Disables the tool with the specified ID.

Syntax

disableTool(toolName)

Parameters

<table>
<thead>
<tr>
<th>toolName</th>
<th>A string identifying the tool.</th>
</tr>
</thead>
</table>

Returns

undefined

enableTool

Enables the tool with the specified ID.
enableTool(toolName)

Parameters

toolName A string identifying the tool.

Returns

undefined

getEventHandler

Obtains the event handler corresponding to the specified index.

Syntax

getEventHandler(index)

Parameters

index An integer identifying the event handler.

Returns

An event handler object.

getRendererName

Obtains the name of the current renderer.

Syntax

getRendererName()

Returns

A string containing the current renderer's name.

ggetView

Acrobat 9.0

Gets the indicated view for the annotation by its index.

See the related method, getView, for setting the view by its index.
**Syntax**
```javascript
getView( index )
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>index</code></td>
<td>The integer index of the view.</td>
</tr>
</tbody>
</table>

**Returns**

View

**getView**

*Acrobat 9.0*

Gets the indicated view for the annotation by its name.

See the related method, `setView`, for setting the view by its name.

**Syntax**
```javascript
getView( name )
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>name</code></td>
<td>The string name of the view.</td>
</tr>
</tbody>
</table>

**Returns**

View

**pause**

*Acrobat 9.0*

Pauses the runtime. This is the same as selecting the Pause toolbar button or menu item.

**Syntax**
```javascript
pause()
```

**Returns**

undefined

**play**

*Acrobat 9.0*
Resumes playback of the runtime. This is the same as selecting the Play toolbar button or menu item.

**Syntax**

```javascript
play()
```

**Returns**

`undefined`

---

**refresh**

Version 7.0.1

Marks the render area dirty so that it is redrawn. This is useful when something changes in the scene but the annotation is in a Loaded and not Live state.

**Syntax**

```javascript
refresh()
```

**Returns**

`undefined`

---

**removeEventHandler**

Unregisters the specified event handler.

**Syntax**

```javascript
removeEventHandler(handler)
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>handler</code></td>
<td>An event handler object representing the event handler.</td>
</tr>
</tbody>
</table>

**Returns**

`undefined`

---

**removeCustomMenuItem**

Removes the custom menu item with the specified ID.

**Syntax**

```javascript
removeCustomMenuItem(menuName)
```
Parameters

menuName   A string identifying the custom menu item.

Returns

undefined

removeCustomToolButton

Removes the custom tool button with the specified ID.

Syntax

removeCustomToolButton(toolName)

Parameters

toolName   A string identifying the custom tool button.

Returns

undefined

setCurrentTool

Sets the current tool to the one with the specified ID.

Syntax

setCurrentTool(toolName)

Parameters

toolName   A string identifying the tool.

Returns

undefined

setCustomMenuItemChecked

Acrobat 7.0.7

Sets the checked state of the provided custom menu item.
Syntax

`setCustomMenuItemChecked( menuItemName, checkedState )`

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>menuItemName</td>
<td>A string identifying the name of the custom menu item.</td>
</tr>
<tr>
<td>checkedState</td>
<td>A Boolean value determining whether the menu should be checked.</td>
</tr>
</tbody>
</table>

Returns

undefined

`setView`

Acrobat 9.0.

Sets the current view for the annotation.

See the related method, `getView`, for getting the view by its index.

Syntax

`setView( index, animate)`

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>index</td>
<td>The integer index of the view to be set.</td>
</tr>
<tr>
<td>animate</td>
<td>(Optional) A Boolean value, when true, indicates that the view should be animated to when set.</td>
</tr>
</tbody>
</table>

Returns

undefined

`setView`

Acrobat 9.0.

Sets the current view for the annotation.

See the related method, `getView`, for getting the view by its name.

Syntax

`setView( name, animate)`
Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>menuItemName</td>
<td>The string name of the view to be set.</td>
</tr>
<tr>
<td>checkedState</td>
<td>(Optional) A Boolean value, when true, indicates that the view should be</td>
</tr>
<tr>
<td></td>
<td>animated to when set.</td>
</tr>
</tbody>
</table>

Returns

undefined
Scene

An object that represents the hierarchy of the 3D related content, including Animation, Light, Material, and Mesh objects. The variable `scene` is a global reference to this object.

Related objects are Animation, Light, Material and Mesh.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ambientIlluminationColor</td>
<td>Color</td>
<td>R</td>
<td>Modulates the ambient Color of all materials.</td>
</tr>
<tr>
<td>animations</td>
<td>SceneObjectList</td>
<td>R</td>
<td>A list of all Animation objects.</td>
</tr>
<tr>
<td>cameras</td>
<td>SceneObjectList</td>
<td>R</td>
<td>A list of all Camera objects in the Scene.</td>
</tr>
<tr>
<td>defaultRenderOptions</td>
<td>RenderOptions</td>
<td>R</td>
<td>A set of all default rendering options for the Scene.</td>
</tr>
</tbody>
</table>
| gridMode                      | string     | R/W    | Acrobat 7.0.7 The display style of the grid that represents a portion of the ground plane in the Scene. It can have one of the following values:
<p>|                               |            |        | ● &quot;off&quot; (no grid)                                                          |
|                               |            |        | ● &quot;wire&quot; (a wireframe grid)                                                |
|                               |            |        | ● &quot;solid&quot; (a solid checkerboard grid)                                      |
|                               |            |        | ● &quot;transparent&quot; (a semi-transparent checkerboard grid)                     |
| GRID_MODE_OFF                 | string     | R      | Acrobat 7.0.7 A string constant for the grid mode of &quot;off&quot;.                 |
| GRID_MODE_SOLID               | string     | R      | Acrobat 7.0.7 A string constant for the grid mode of &quot;solid&quot;.               |
| GRID_MODE_TRANSPARENT         | string     | R      | Acrobat 7.0.7 A string constant for the grid mode of &quot;transparent&quot;.         |</p>
<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRID_MODE_WIRE</td>
<td>string</td>
<td>R</td>
<td>A string constant for the grid mode of &quot;wire&quot;.</td>
</tr>
<tr>
<td>gridSize</td>
<td>number</td>
<td>R</td>
<td>The number of squares on the ground plane grid.</td>
</tr>
<tr>
<td>lengthUnits</td>
<td>number</td>
<td>R</td>
<td>The scale of a unit of length, specified in meters.</td>
</tr>
<tr>
<td>LIGHT_MODE_FILE</td>
<td>string</td>
<td>R</td>
<td>A string constant for the light mode of &quot;file&quot;.</td>
</tr>
<tr>
<td>LIGHT_MODE_NONE</td>
<td>string</td>
<td>R</td>
<td>A string constant for the light mode of &quot;none&quot;.</td>
</tr>
<tr>
<td>LIGHT_MODE_WHITE</td>
<td>string</td>
<td>R</td>
<td>A string constant for the light mode of &quot;white&quot;.</td>
</tr>
<tr>
<td>LIGHT_MODE_DAY</td>
<td>string</td>
<td>R</td>
<td>A string constant for the light mode of &quot;day&quot;.</td>
</tr>
<tr>
<td>LIGHT_MODE_BRIGHT</td>
<td>string</td>
<td>R</td>
<td>A string constant for the light mode of &quot;bright&quot;.</td>
</tr>
<tr>
<td>LIGHT_MODE_RGB</td>
<td>string</td>
<td>R</td>
<td>A string constant for the light mode of &quot;rgb&quot;.</td>
</tr>
<tr>
<td>LIGHT_MODE_NIGHT</td>
<td>string</td>
<td>R</td>
<td>A string constant for the light mode of &quot;night&quot;.</td>
</tr>
<tr>
<td>LIGHT_MODE_BLUE</td>
<td>string</td>
<td>R</td>
<td>A string constant for the light mode of &quot;blue&quot;.</td>
</tr>
<tr>
<td>LIGHT_MODE_RED</td>
<td>string</td>
<td>R</td>
<td>A string constant for the light mode of &quot;red&quot;.</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Access</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LIGHT_MODE_CUBE</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the light mode of &quot;cube&quot;.</td>
</tr>
<tr>
<td>LIGHT_MODE_CAD</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the light mode of &quot;cad&quot;.</td>
</tr>
<tr>
<td>LIGHT_MODE_HEADLAMP</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the light mode of &quot;headlamp&quot;.</td>
</tr>
<tr>
<td>lights</td>
<td>SceneObjectList</td>
<td>R</td>
<td>A list of all Light objects in the Scene.</td>
</tr>
<tr>
<td>lightScaleFactor</td>
<td>number</td>
<td>R/W</td>
<td>A uniform scale factor for all Light objects in the Scene.</td>
</tr>
<tr>
<td>lightScheme</td>
<td>string</td>
<td>R/W</td>
<td>Acrobat 7.0.7 The current, preconfigured lighting scheme for the Scene.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>It can take one of the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;file&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;none&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;white&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;day&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;bright&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;rgb&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;night&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;blue&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;red&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;cube&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;cad&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;headlamp&quot;</td>
</tr>
<tr>
<td>materials</td>
<td>SceneObjectList</td>
<td>R</td>
<td>A list of all Material objects.</td>
</tr>
<tr>
<td>meshes</td>
<td>SceneObjectList</td>
<td>R</td>
<td>A list of all Mesh objects in the Scene.</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Access</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>nodes</td>
<td>SceneObjectList</td>
<td>R</td>
<td>A list of all Node objects except the default Camera and default Light objects.</td>
</tr>
<tr>
<td>outlineAngle</td>
<td>number</td>
<td>R/W</td>
<td>Acrobat 7.0.7 The crease angle (in degrees) for the appearance of lines in Illustration render modes.</td>
</tr>
<tr>
<td>showGrid</td>
<td>Boolean</td>
<td>R/W</td>
<td>Acrobat 7.0.7 Determines whether the ground plane grid is displayed.</td>
</tr>
<tr>
<td>renderDoubleSided</td>
<td>Boolean</td>
<td>R/W</td>
<td>Acrobat 8.1 Toggles if back-facing polygons are rendered.</td>
</tr>
<tr>
<td>renderMode</td>
<td>string</td>
<td>R/W</td>
<td>Acrobat 7.0.7 The default rendering type for all objects in the Scene, which can be one of the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;default&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;bounding box&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;transparent bounding box&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;transparent bounding box outline&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;vertices&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;shaded vertices&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;wireframe&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;shaded wireframe&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;solid&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;transparent&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;solid wireframe&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;transparent wireframe&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;illustration&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;solid outline&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;shaded illustration&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● &quot;hidden wireframe&quot;</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Access</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>----------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RENDER_MODE_DEFAULT</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the render mode of &quot;default&quot;.</td>
</tr>
<tr>
<td>RENDER_MODE_BOUNDING_BOX</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the render mode of &quot;bounding box&quot;.</td>
</tr>
<tr>
<td>RENDER_MODE_TRANSPARENT_BOUNDING_BOX</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the render mode of &quot;transparent bounding box&quot;.</td>
</tr>
<tr>
<td>RENDER_MODE_TRANSPARENT_BOUNDING_BOX_OUTLINE</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the render mode of &quot;transparent bounding box outline&quot;.</td>
</tr>
<tr>
<td>RENDER_MODE_VERTICES</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the render mode of &quot;vertices&quot;.</td>
</tr>
<tr>
<td>RENDER_MODE_SHADED_VERTICES</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the render mode of &quot;shaded vertices&quot;.</td>
</tr>
<tr>
<td>RENDER_MODE_WIREFRAME</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the render mode of &quot;wireframe&quot;.</td>
</tr>
<tr>
<td>RENDER_MODE_SHADED_WIREFRAME</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the render mode of &quot;shaded wireframe&quot;.</td>
</tr>
<tr>
<td>RENDER_MODE_SOLID</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the render mode of &quot;solid&quot;.</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Access</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RENDER_MODE_TRANSPARENT</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the render mode of &quot;transparent&quot;.</td>
</tr>
<tr>
<td>RENDER_MODE_SOLID_WIREFRAME</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the render mode of &quot;solid wireframe&quot;.</td>
</tr>
<tr>
<td>RENDER_MODE_TRANSPARENT_WIREFRAME</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the render mode of &quot;transparent wireframe&quot;.</td>
</tr>
<tr>
<td>RENDER_MODE_ILLUSTRATION</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the render mode of &quot;illustration&quot;.</td>
</tr>
<tr>
<td>RENDER_MODE_SOLID_OUTLINE</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the render mode of &quot;solid outline&quot;.</td>
</tr>
<tr>
<td>RENDER_MODE_SHADED_ILLUSTRATION</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the render mode of &quot;shaded illustration&quot;.</td>
</tr>
<tr>
<td>RENDER_MODE_HIDDEN_WIREFRAME</td>
<td>string</td>
<td>R</td>
<td>Acrobat 7.0.7 A string constant for the render mode of &quot;hidden wireframe&quot;.</td>
</tr>
<tr>
<td>selectedNode</td>
<td>Node</td>
<td>R/W</td>
<td>Acrobat 8.1 The currently selected Node.</td>
</tr>
<tr>
<td>showAxes</td>
<td>Boolean</td>
<td>R/W</td>
<td>Acrobat 7.0.7 Determines whether the world axes are displayed.</td>
</tr>
<tr>
<td>showOrientationAxes</td>
<td>Boolean</td>
<td>R/W</td>
<td>Acrobat 9.0 Determines whether the orientation axes are displayed.</td>
</tr>
</tbody>
</table>
activateAnimation

Sets the given Animation to be active on its Node objects.

Syntax

activateAnimation(animation)

Parameters

animation The Animation object to be activated.

Returns

undefined

addFlashForeground

Acrobat 9.0

Adds the provided FlashMovie as a foreground element within the 3D scene.

Syntax

addFlashForeground(movie)

Parameters

movie The FlashMovie to be added as a foreground element.

Property | Type | Access | Description
---|---|---|---
smoothing | Boolean | R/W | Acrobat 7.0.7
smoothingAngle | number | R/W | Acrobat 7.0.7
smoothingOverride | Boolean | R/W | Acrobat 7.0.7

When true, smoothing is enabled for meshes in the scene.
The default smoothing angle (in degrees) for meshes in the scene.
When set to true, overrides the smoothing values from the loaded model file.
addModel

Adds a model Resource to the top level of the Scene.

Syntax

```
addModel(modelRes)
```

Parameters

- `modelRes` The Resource object to be added.

Returns

A Node object representing the top-level Mesh of the loaded model.

createClippingPlane

Creates a new clipping plane.

Syntax

```
createClippingPlane()
```

Returns

A ClippingPlane object.

cREATELIGHT

Creates a new Light and attaches it to the Scene.

Syntax

```
createLight()
```

Returns

A Light object.

cREATE SquareMesh

Creates a Mesh that is a unit square. The default UV parameterization fits once in U and V.
Syntax

createSquareMesh(sizeX, sizeY, name)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sizeX</td>
<td>Model units in the x direction used to size the Mesh.</td>
</tr>
<tr>
<td>sizeY</td>
<td>Model units in the y direction used to size the Mesh.</td>
</tr>
<tr>
<td>name</td>
<td>(Optional) The name that is assigned to the Mesh.</td>
</tr>
</tbody>
</table>

Returns

A Mesh object.

computeBoundingBox

Computes the BoundingBox of the Scene.

Syntax

computeBoundingBox()

Returns

A BoundingBox object.

update

Applies all changes to the Scene.

Syntax

update()

Returns

undefined
SceneObject

The base type for objects within the Scene, including Animation, Material, and Node objects.

Related objects are Scene, Animation, Light, Material, and Mesh.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string</td>
<td>The name of the SceneObject object.</td>
</tr>
<tr>
<td>objectGUID</td>
<td>string</td>
<td>Deprecated A value that uniquely identifies the SceneObject in custom data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This property has a default value.</td>
</tr>
<tr>
<td>objectID</td>
<td>integer</td>
<td>An unsigned 32-bit value that uniquely identifies the SceneObject. This property can be assigned, but it does not have a default value. It always returns 0.</td>
</tr>
</tbody>
</table>
SceneObjectList

A structure that contains references to several SceneObject objects.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>integer</td>
<td>R</td>
<td>The number of elements in the SceneObjectList.</td>
</tr>
</tbody>
</table>

getByGUID

Deprecated

Obtains the specified SceneObject object from the list.

Syntax

getByGUID(guid)

Parameters

| guid | A string representing the GUID for the specified element. |

Returns

A SceneObject object.

getByID

Obtains the specified SceneObject object from the list.

Syntax

getByID(id)

Parameters

| id | An integer representing the object identifier for the specified SceneObject object. |

Returns

A SceneObject object.

getByIndex

Obtains the specified SceneObject object from the list.
Syntax
getByIndex(index)

Parameters

index  An integer representing the index of the specified SceneObject object.

Returns
A SceneObject object.

getByName

Obtains the specified SceneObject object from the list.

Syntax
getByName(name)

Parameters

name  A string representing the name of the specified SceneObject object.

Returns
A SceneObject object.

removeAll

Deprecated

Removes all the SceneObject objects from the list, but does not delete them from the Scene.

Syntax
removeAll()

Returns
undefined

removeByIndex

Deprecated

Removes the specified SceneObject object from the list, but does not delete it from the Scene.
Syntax

removeByIndex(index)

Parameters

index An index to the specified element.

Returns

undefined

removeItem

Deprecated

Removes a SceneObject object from the list, but does not delete it from the Scene.

Syntax

removeItem(scene)

Parameters

scene A scene object that is to be removed.

Returns

undefined
ScrollWheelEvent

(Acrobat 8.1) An object that is passed as an argument to the `onEvent` method of the `ScrollWheelEventHandler` object. A `ScrollWheelEvent` object is created when the mouse scroll wheel is activated over an active 3D `Canvas` object.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>canvas</td>
<td>Canvas</td>
<td>R</td>
<td>The Canvas in which the <code>ScrollWheelEvent</code> took place.</td>
</tr>
<tr>
<td>canvasPixelHeight</td>
<td>integer</td>
<td>R</td>
<td>The height, measured in pixels, of the Canvas in which the <code>ScrollWheelEvent</code> took place.</td>
</tr>
<tr>
<td>canvasPixelWidth</td>
<td>integer</td>
<td>R</td>
<td>The width, measured in pixels, of the Canvas in which the <code>ScrollWheelEvent</code> took place.</td>
</tr>
<tr>
<td>ctrlKeyDown</td>
<td>Boolean</td>
<td>R</td>
<td>Determines whether the Ctrl key (Windows) or Command key (Mac OS) was pressed.</td>
</tr>
<tr>
<td>currentTool</td>
<td>string</td>
<td>R</td>
<td>The name of the current tool.</td>
</tr>
<tr>
<td>deltaY</td>
<td>number</td>
<td>R</td>
<td>The amount the scroll wheel was moved in the Y direction.</td>
</tr>
<tr>
<td>shiftKeyDown</td>
<td>Boolean</td>
<td>R</td>
<td>Determines whether the Shift key was pressed.</td>
</tr>
</tbody>
</table>
ScrollWheelEventHandler

(Acrobat 8.1) An object that exposes a callback mechanism that allows a function to be evaluated when an event occurs. Event handlers are registered with the Runtime method `addEventHandler`.

**ScrollWheelEventHandler**

A constructor that creates a new `ScrollWheelEventHandler`.

**Syntax**

```javascript
new ScrollWheelEventHandler()
```

**Returns**

A `ScrollWheelEventHandler` object.

**onEvent**

A method that is called when the scroll wheel is used in an active 3D annotation.

**Syntax**

```javascript
onEvent(event)
```

**Parameters**

| event       | A `ScrollWheelEvent` object representing the event. |

**Returns**

undefined
SelectionEvent

(Acrobat 8.1) An object that is passed as an argument to the `onEvent` method of the `SelectionEventHandler` object.

A `SelectionEvent` object is created when an object is selected from an active 3D Canvas object or from a model tree. If the selection is made from a Canvas object, a `MouseEvent` is also created.

**Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td>Node</td>
<td>R</td>
<td>The Node that is the target of the selection change.</td>
</tr>
<tr>
<td>selected</td>
<td>Boolean</td>
<td>R</td>
<td>The selected state of the target Node.</td>
</tr>
</tbody>
</table>
SelectionEventHandler

(Acrobat 8.1) An object that exposes a callback mechanism that allows a function to be evaluated when an event occurs. Event handlers are registered with the Runtime method addEventHandler.

**SelectionEventHandler**

A constructor that creates a new SelectionEventHandler object.

**Syntax**

```javascript
new SelectionEventHandler()
```

**Returns**

A SelectionEventHandler object.

**onEvent**

A method that is called when the selection state changes in an active 3D annotation.

**Syntax**

```javascript
onEvent(event)
```

**Parameters**

<table>
<thead>
<tr>
<th>parameter</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>event</td>
<td>A ScrollWheelEvent object representing the event.</td>
</tr>
</tbody>
</table>

**Returns**

undefined
StateEvent

Acrobat 9.0

An object that is passed as an argument to the `onEvent` method of the `StateEventHandler` object. A `StateEvent` object is created when state data must be stored or loaded for the scene, such as when a new comment view is created or invoked for the annotation.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stateString</td>
<td>string</td>
<td>R</td>
<td>If the <code>SaveEvent</code> type is &quot;load&quot;, this property contains the state data that was stored as part of the corresponding &quot;save&quot; <code>StateEvent</code>. If the <code>SaveEvent</code> type is &quot;save&quot;, the <code>stateString</code> is undefined.</td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td>R</td>
<td>The type of <code>StateEvent</code>, this property has a value of either &quot;load&quot; or &quot;save&quot;.</td>
</tr>
<tr>
<td>TYPE_LOAD</td>
<td>string</td>
<td>R</td>
<td>A string constant for the <code>StateEvent</code> type of &quot;load&quot;. The state data that was stored as part of the original <code>stateEvent</code>.</td>
</tr>
<tr>
<td>TYPE_SAVE</td>
<td>string</td>
<td>R</td>
<td>A string constant for the <code>StateEvent</code> type of &quot;save&quot;.</td>
</tr>
</tbody>
</table>
StateEventHandler

Acrobat 9.0

An object that exposes a callback mechanism that allows a function to be evaluated when a state event occurs. Event handlers are registered with the `Runtime` method `addEventHandler`.

onEvent

A method that is called when state data must be stored or loaded for the annotation. The return value is stored as the `stateString` for the given `StateEvent`.

Syntax

```
onEvent(event)
```

Parameters

- `event` - A `StateEvent` object representing the event.

Returns

- string or undefined

StateEventHandler

The constructor that creates a new `StateEventHandler`.

Syntax

```
new StateEventHandler()
```

Returns

- A `StateEventHandler` object.
Texture

A `Texture` object represents the mapping of a texture. All `Texture` properties have read-write permissions.

### Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>amount</td>
<td>number</td>
<td>The degree to which the <code>Texture</code> is applied, which can be a value from 0.0 to 1.0.</td>
</tr>
<tr>
<td>angle</td>
<td>number</td>
<td>The rotation angle, measured in degrees, of the map.</td>
</tr>
<tr>
<td>clampU</td>
<td>Boolean</td>
<td>Determines whether the map should be clamped in the U direction.</td>
</tr>
<tr>
<td>clampV</td>
<td>Boolean</td>
<td>Determines whether the map should be clamped in the V direction.</td>
</tr>
<tr>
<td>image</td>
<td>Image</td>
<td>Acrobat 7.0.7 The <code>Image</code> to be used by the <code>Texture</code>.</td>
</tr>
<tr>
<td>modulate</td>
<td>Boolean</td>
<td>Determines whether to set the blend mode of the <code>Texture</code> with its corresponding color.</td>
</tr>
<tr>
<td>uOffset</td>
<td>number</td>
<td>The U-offset of the given map.</td>
</tr>
<tr>
<td>uScale</td>
<td>number</td>
<td>The U-scale of the given map.</td>
</tr>
<tr>
<td>use3DSStyleMapping</td>
<td>Boolean</td>
<td>Determines whether to use 3D Studio style map parameterizations.</td>
</tr>
<tr>
<td>vOffset</td>
<td>number</td>
<td>The V-offset of the given map.</td>
</tr>
<tr>
<td>vScale</td>
<td>number</td>
<td>The V-scale of the given map.</td>
</tr>
</tbody>
</table>

**getImage**

*Deprecated*

Gets the `Image` currently used by the `Texture`.

**Syntax**

`getImage()`

**Returns**

The `Image` currently being used.
setImage

Deprecated

Sets the Image to be used by the Texture.

Syntax

```
setImage(image)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>image</td>
<td>The Image to be used.</td>
</tr>
</tbody>
</table>

Returns

undefined
TimeEvent

An object that is passed as an argument to the `TimeEventHandler` object's `onEvent` method.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>deltaTime</td>
<td>number</td>
<td>R</td>
<td>The difference between the current time and the last time.</td>
</tr>
<tr>
<td>time</td>
<td>number</td>
<td>R</td>
<td>The current time.</td>
</tr>
</tbody>
</table>
TimeEventHandler

An object that exposes a callback mechanism that allows a function to be evaluated when an event occurs. Event handlers are registered with the Runtime `addEventHandler` method.

**TimeEventHandler**

A constructor that creates a new `TimeEventHandler` object.

**Syntax**

```
new TimeEventHandler()
```

**Returns**

A `TimeEventHandler` object.

**onEvent**

A method that is called when simulation time is incremented in an active 3D annotation.

**Syntax**

```
onEvent(event)
```

**Parameters**

- `event` A `TimeEvent` object representing the event.

**Returns**

undefined
ToolEvent

An object that is passed as an argument to the onEvent method of the ToolEventHandler object.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>canvas</td>
<td>Canvas</td>
<td>R</td>
<td>The Canvas that is the target of the ToolEvent.</td>
</tr>
<tr>
<td>canvasPixelHeight</td>
<td>integer</td>
<td>R</td>
<td>The height, measured in pixels, of the Canvas for which the ToolEvent is intended.</td>
</tr>
<tr>
<td>canvasPixelWidth</td>
<td>integer</td>
<td>R</td>
<td>The width, measured in pixels, of the Canvas for which the ToolEvent is intended.</td>
</tr>
<tr>
<td>currentTool</td>
<td>string</td>
<td>R</td>
<td>The name of the current tool.</td>
</tr>
<tr>
<td>toolName</td>
<td>string</td>
<td>R</td>
<td>The name of the tool that was clicked.</td>
</tr>
</tbody>
</table>
ToolEventHandler

An object that exposes a callback mechanism that allows a function to be evaluated when an event occurs. Event handlers are registered with the Runtime `addEventHandler` method.

**ToolEventHandler**

A constructor that creates a new `ToolEventHandler` object.

**Syntax**

```javascript
new ToolEventHandler()
```

**Returns**

A `ToolEventHandler` object.

**onEvent**

A method that is called when a tool button is pressed on the 3D toolbar.

**Syntax**

```javascript
onEvent(event)
```

**Parameters**

- `event` A `ToolEvent` object representing the event.

**Returns**

undefined
Vector3

An object comprised of three values that represent a point in space or a direction and magnitude.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>number</td>
<td>R/W</td>
<td>The x component of the Vector3 object.</td>
</tr>
<tr>
<td>y</td>
<td>number</td>
<td>R/W</td>
<td>The y component of the Vector3 object.</td>
</tr>
<tr>
<td>z</td>
<td>number</td>
<td>R/W</td>
<td>The z component of the Vector3 object.</td>
</tr>
<tr>
<td>length</td>
<td>number</td>
<td>R</td>
<td>The length of the Vector3 object.</td>
</tr>
</tbody>
</table>

Vector3

A constructor that initializes the new object to \((0.0, 0.0, 0.0)\).

Syntax

```
new Vector3()
```

Returns

A Vector3 object.

Vector3

A constructor that initializes the new object to the specified components.

Syntax

```
new Vector3(x, y, z)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>The x component used to initialize the new object.</td>
</tr>
<tr>
<td>y</td>
<td>The y component used to initialize the new object.</td>
</tr>
<tr>
<td>z</td>
<td>The z component used to initialize the new object.</td>
</tr>
</tbody>
</table>

Returns

A Vector3 object.
add

Adds the specified Vector3 to the current one.

Syntax
add(offset)

Parameters

| offset | The Vector3 object to be added to the current one. |

Returns

A Vector3 object.

addInPlace

Adds the specified Vector3 to the current one, and updates the current Vector3 with the resulting value.

Syntax
addInPlace(offset)

Parameters

| offset | The Vector3 object to be added to the current one. |

Returns

undefined

addScaled

Adds the specified Vector3 with the scaled offset to the current one.

Syntax
addScaled(offset, scale)

Parameters

| offset  | The Vector3 object to be added to the current one. |
| scale   | The scaling factor for the offset. |
Returns

A Vector3 object.

addScaledInPlace

Adds the specified Vector3 with the scaled offset to the current one, and updates the current Vector3 with the resulting value.

Syntax

addScaledInPlace(offset, scale)

Parameters

<table>
<thead>
<tr>
<th>offset</th>
<th>The Vector3 object to be added to the current one.</th>
</tr>
</thead>
<tbody>
<tr>
<td>scale</td>
<td>The scaling factor for the offset.</td>
</tr>
</tbody>
</table>

Returns

undefined

blend

Blends the current and specified Vector3 by the specified degree.

Syntax

blend(vec, blendFactor)

Parameters

<table>
<thead>
<tr>
<th>vec</th>
<th>The Vector3 object to be blended with the current one.</th>
</tr>
</thead>
<tbody>
<tr>
<td>blendFactor</td>
<td>The degree of blending to be applied, which can be a value from 0.0 to 1.0.</td>
</tr>
</tbody>
</table>

Returns

A Vector3 object.

blendInPlace

Blends the current and specified Vector3 by the specified degree, and updates the current Vector3 with the resulting value.

Syntax

blendInPlace(vec, blendFactor)
Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vec</td>
<td>The Vector3 object to be blended with the current one.</td>
</tr>
<tr>
<td>blendFactor</td>
<td>The degree of blending to be applied, which can be a value from 0.0 to 1.0.</td>
</tr>
</tbody>
</table>

Returns

undefined

cross

Calculates the cross product between the specified Vector3 and the current one.

Syntax

cross(vec)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vec</td>
<td>The Vector3 object to be used in calculating the cross product.</td>
</tr>
</tbody>
</table>

Returns

A Vector3 object.

dot

Calculates the dot product between the specified Vector3 and the current one.

Syntax

dot(vec)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vec</td>
<td>The Vector3 object to be used in calculating the dot product.</td>
</tr>
</tbody>
</table>

Returns

A number value representing the scalar dot product.

normalize

Normalizes the current Vector3.
Syntax
normalize()

Returns
undefined

scale

Scales the current Vector3.

Syntax
scale(scale)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>scale</td>
<td>A number value used to scale the current Vector3.</td>
</tr>
</tbody>
</table>

Returns
A Vector3 object.

scaleInPlace

Scales the current Vector3, and updates the current Vector3 with the resulting value.

Syntax
scaleInPlace(scale)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>scale</td>
<td>A number value used to scale the current Vector3.</td>
</tr>
</tbody>
</table>

Returns
undefined

set

Sets the current Vector3 to the value contained in the specified Vector3.

Syntax
set(vec)
### Parameters

| vec | The Vector3 used to set the current Vector3. |

### Returns

undefined

### set

Acrobat 7.0.7

Sets the current Vector3 to the values contained in the specified components.

#### Syntax

```javascript
set(x, y, z)
```

#### Parameters

<table>
<thead>
<tr>
<th>x</th>
<th>The x component used to set the current Vector3.</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>The y component used to set the current Vector3.</td>
</tr>
<tr>
<td>z</td>
<td>The z component used to set the current Vector3.</td>
</tr>
</tbody>
</table>

### Returns

undefined

### set3

Deprecated

Sets the current Vector3 to the values contained in the specified components.

#### Syntax

```javascript
set3(x, y, z)
```

#### Parameters

<table>
<thead>
<tr>
<th>x</th>
<th>The x component used to set the current Vector3.</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>The y component used to set the current Vector3.</td>
</tr>
<tr>
<td>z</td>
<td>The z component used to set the current Vector3.</td>
</tr>
</tbody>
</table>
Returns
undefined

subtract

Subtracts the specified Vector3 from the current one.

Syntax
subtract(offset)

Parameters

| offset | The Vector3 object to be subtracted from the current one. |

Returns

A Vector3 object.

subtractInPlace

Subtracts the specified Vector3 from the current one, and updates the current Vector3 with the resulting value.

Syntax
subtractInPlace(offset)

Parameters

| offset | The Vector3 object to be subtracted from the current one. |

Returns
undefined
View

Acrobat 9.0

An object that represents a named view for the annotation.

See the `viewCount` property and the `getView` and `setView` methods of the `Runtime` object.

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string</td>
<td>R</td>
<td>The name of the view.</td>
</tr>
</tbody>
</table>
New Features and Changes

This chapter summarizes the new features and changes introduced in Acrobat.

Acrobat X changes

There are no changes to JavaScript for Acrobat 3D Annotations in Acrobat X.

Acrobat 9.0 changes

This section describes the changes introduced in Acrobat 9 Pro Extended.

New objects

The following new objects are introduced to support Rich Media annotations and Geospatial features in Acrobat 9.0:

- **FlashEvent** object: This new object has these properties: `command`, `target`, `type`, `TYPE_COMMAND`, `TYPE_PROGRESS`, `TYPE_STATECHANGE`, and `value`.
- **FlashEventHandler** object: This new object has one property, `target`, and these methods: `onEvent` and `FlashEventHandler`.
- **FlashMovie** object: This new object has these properties: `alignMode`, `ALIGN_MODE_BOTTOM`, `ALIGN_MODE_LEFT`, `ALIGN_MODE_RIGHT`, `ALIGN_MODE_TOP`, `backgroundColor`, `desiredResolutionX`, `desiredResolutionY`, `frameNum`, `hitEnabled`, `id`, `loop`, `opacity`, `percentLoaded`, `playing`, `quality`, `readyState`, `resolutionType`, `RESOLUTION_TYPE_CUSTOM`, `RESOLUTION_TYPE_MOVIE`, `RESOLUTION_TYPE_WINDOW`, `scaleMode`, `SCALE_MODE_EXACT_FIT`, `SCALE_MODE_NO_BORDER`, `SCALE_MODE_SHOW_ALL`, `totalFrames`, `x`, and `y`. It also has these methods: `FlashMovie`, `call`, `getVariable`, `gotoFrame`, `isPlaying`, `pan`, `play`, `rewind`, `setVariable`, `setZoomRect`, `stop`, and `zoom`.
- **StateEvent** object: This new object has these properties: `stateString`, `type`, `TYPE_LOAD`, and `TYPE_SAVE`.
- **StateEventHandler** object: This new object has these methods: `onEvent` and `StateEventHandler`.
- **View** object: This new object has the `name` property.

Additional properties and methods in existing objects

- The **Background** object has one additional property, `FlashMovie`.
- The **Resource** object has additional properties: `type` (a new value of "flash") and `TYPE_FLASH`.
- This Material object has one additional method, `attachFlashMovie`.
- The **Runtime** object has additional properties: `overrideFlyTool`, `TOOL_NAME_FLY`, and `viewCount`. It also has additional methods: `pause`, `play`, `getView`, and `setView`.
- The **Scene** object has a new property, `showOrientationAxes`, and a new method, `addFlashForeground`.
APIs for versions earlier than 9.0

The following properties and methods were available but undocumented in earlier versions of this reference.

- Two properties were defined for the `Camera` object in version 7, `absoluteBindingScale` and `useAbsoluteBinding`.
- Two `set` methods were defined for the `Matrix4x4` object in version 8.1, one takes an array argument and the other a list of numbers.
- The `setCustomMenuItemChecked` method was defined for the `Runtime` object in version 7.0.7.
- The `opacity` property was defined for the `Node` object in version 7.0.7.

API changes

- The `computeBoundingBox` method is documented as a method of the `Mesh` object, not the `Node` object, as previously published.
- The property previously documented for the `Runtime` object as `overrideWheelSpeed` was incorrect. It is now properly identified as `overrideScrollWheel`, and the description changed.
- The `RENDER_MODE_SOLID_WIREFRAME` property for the `Scene` object was incorrectly documented as `RENDER_MODE_SHADED_SOLID_WIREFRAME`.
- Deprecate the methods `removeAll`, `removeByIndex`, and `removeItem` for the `SceneObjectList` object. Their descriptions were modified to indicate that these methods remove elements from the `SceneObjectList`, but not from the `Scene`.

Acrobat 8.1 changes

This section describes the changes introduced in Acrobat 8.1.

New objects

The following objects are new: `ScrollWheelEvent`, `ScrollWheelEventHandler`, `SelectionEvent`, and `SelectionEventHandler`.

Additional properties in existing objects

The `HitInfo` object has additional properties: `material`, `surfaceNormal`, and `textureCoordinate`.

The `Node` object has an additional property: `metadataString`.

The `Light` object has an additional property: `directionLocal` (Acrobat 7, but previously undocumented).

The `Runtime` object has additional properties: `canvasCount`, `overrideSpinTool`, `scrollWheelSpeed`, `speedThreshold`, `strafeSpeed`, and `walkSpeed`.

The `Scene` object has additional properties: `node` and `selected`. 
Deprecated objects or properties

The following APIs were deprecated:

- `CameraEvent.isNewCanvas` (a property)
- `Dummy` (an object)
- `Procedural` (an object)
- `SceneObject.objectGUID` (a property)
- `SceneObject.getByGUID` (a method)

Acrobat 8.0 changes

This section describes the changes introduced in Acrobat 8.0.

Additional properties in existing objects

The `Runtime` object has additional properties: `overrideSpinTool` and `TOOL_NAME_SPIN`. 
Index

Numbers
3D JavaScript engine 9
   accessing 9
   accessing using the SceneContext3D object 9

A
absoluteBindingScale property 21
accessing the 3D JavaScript engine 9
   using the SceneContext3D object 9
activateAnimation method 97
add method 117
addCustomMenuItem method 83
addCustomToolButton method 83
addEventHandler method 84
addFlashForeground method 97
addInPlace method 117
addModel method 98
addScaled method 117
addScaledInPlace method 118
ALIGN_MODE_BOTTOM property 35
ALIGN_MODE_LEFT property 35
ALIGN_MODE_RIGHT property 35
alignMode property 35
ambientColor property 49
ambientIlluminationColor property 91
ambientTexture property 49
amount property 110
angle property 110
Animation object
   about 16
   currentTime property 16
   endTime property 16
   framesPerSecond property 16
   length property 16
   startTime property 16
   animations property 91
Annot3D.context3D property 9
ATTENUATION_ABC property 47
ATTENUATION_NONE property 47
attenuationA property 47
attenuationB property 47
attenuationC property 47
attenuationType property 47

B
Background object
   about 17
   FlashMovie property 17
   getColor method 17
   getImage method 17
   image property 17
   setColor method 17
   setImage method 18
   background property 26
   binding property 21, 24
   BINDING_HORIZONTAL property 21
   BINDING_MAX property 21
   BINDING_MIN property 21
   BINDING_VERTICAL property 21
   blend method 118
   blendInPlace method 118
   Bone object 19
   BoundingBox object
      about 20
      center property 20
      max property 20
      min property 20
      boundingBoxColor property 77
      brightness property 47
      bumpTexture property 49
      BUTTON_TYPE_PUSH property 80
      BUTTON_TYPE_TOOL property 80

C
call method 37
Camera object
   about 21
   absoluteBindingScale property 21
   binding property 21
   BINDING_HORIZONTAL property 21
   BINDING_MAX property 21
   BINDING_MIN property 21
   BINDING_VERTICAL property 21
   far property 21
   fov property 21
   getDirectionFromScreen method 23
   getScreenFromPosition method 22
   near property 21
   position property 21
   positionLocal property 21
   projectionType property 22
   roll property 22
   target property 22
   targetPosition property 22
   targetPositionLocal property 22
   TYPE_ORTHORAPHIC property 22
   TYPE_PERSPECTIVE property 22
   up property 22
   upLocal property 22
   viewPlaneSize property 22
   CameraEvent object
      about 24
      binding property 24
      canvas property 24
      currentTool property 24
far property 24
fov property 24
isNewCanvas property 24
near property 24
projectionType property 24
targetDistance property 24
transform property 24
viewPlaneSize property 24
CameraEventHandler method 25
CameraEventHandler object
about 25
Cameras property 91
Canvas object
about 26
background property 26
camera method 26
setCamera method 26
canvas property 24, 44, 62, 65, 75, 76, 104, 114
canvasCount property 80
canvasPixelHeight property 44, 65, 75, 104, 114
canvasPixelWidth property 44, 65, 75, 104, 114
center property 20
characterCode property 44
clampU property 110
clampV property 110
ClippingPlane object
about 27
remove method 27
clippingPlaneColor property 77
clippingPlaneIntersectionColor property 77
Color method 28
Color object
about 28
Color method 28
r, g, b properties 28
set method 28
set3 method 29
color property 47
command property 33
computeBoundingBox method 64, 99
Console object
about 31
print method 31
println method 31
corrections made in Acrobat 9.0 125
count property 101
createClippingPlane method 98
createLight method 98
createSquareMesh method 98
cross method 119
ctrlKeyDown property 45, 65, 80, 104
currentTime property 16
currentTool property 24, 45, 62, 65, 75, 104, 114

defaultAmbientColor property 77
defaultDiffuseColor property 77
defaultEmissiveColor property 77
defaultRenderOptions property 91
defaultSpecularColor property 91
deltaTime property 112
deltaY property 104
desiredResolutionY property 35
detachFromCurrentAnimation method 70
determinant property 51
diffuseColor property 49
diffuseTexture property 49
direction property 49
directionLocal property 48
disableTool method 84
distance property 42
dot method 119
Dummy object 32

E
emissiveColor property 49
emissiveTexture property 49
enableTool method 84
desiredResolutionY property 35
target property 34
FlashMovie method 37
FlashMovie object
about 35
FlashMovie method 37
ALIGN_MODE_BOTTOM property 35
ALIGN_MODE_LEFT property 35
ALIGN_MODE_RIGHT property 35
alignMode Property 35
call method 37
desiredResolutionY property 35
FlashMovie method 37
frameNum property 35
getVariable method 38
gotoFrame method 38
hitEnabled property 35
id property 36
isPlaying method 39
loop property 36
opacity property 36
play method 39
playing property 36
quality property 36
readyState property 36
resolutionType property 36
RESOLUTION_TYPE_CUSTOM property 36
RESOLUTION_TYPE.Movie property 36
RESOLUTION_TYPE_WINDOW property 36
rewind method 39
SCALE_Mode.EXACT_FIT property 36
SCALE_MODE_NO_BORDER property 36
SCALE_MODE_SHOW_ALL property 37
scaleMode property 36
setVariable method 40
setZoomRect 40
stop method 41
totalFrames property 37
x property 37
y property 37
zoom method 41
FlashMovie property 17
Flashovers object
about 33
command property 33
target property 33
type property 33
TYPE_COMMAND property 33
TYPE_PROGRESS property 33
TYPE_STATECHANGE property 33
value property 33
fov property 21, 24
frameNum property 35
framesPerSecond property 16
G
getByGUID method 101
getById method 101
getIndex method 101
getByName method 102
getCamera method 26
getColor method 17
getAddressFromScreen method 23
getEventHandler method 85
getImage method 17, 110
getRendererName method 85
getScreenFromPosition method 22
getVariable method 38
getView method 85, 86
gotoFrame method 38
GRID_MODE_OFF property 91
GRID_MODE_SOLID property 91
GRID_MODE_TRANSPARENT property 91
GRID_MODE_WIRE property 92
gridMode property 91
gridSize property 92
H
height property 43
hitEnabled property 35, 69
HitInfo object
about 42
distance property 42
material property 42
position property 42

surfaceNormal property 42
target property 42
textureCoordinate property 42
hits property 65
Host object 42
I
id property 36
illustrationRenderModeFaceColor property 77
illustrationRenderModeLineColor property 77
Image method 43
Image object
about 43
height property 43
Image method 43
width property 43
image property 17, 110
info property 69
innerConeAngle property 48
innerRadius property 48
instances property 80
interpolate method 73
interpolateInPlace method 73
inverse property 51
invertInPlace method 52
isDoubleClick property 65
isEqual method 52
isMouseDown property 65
isMouseHit property 65
isMouseMove property 65
isMouseOut property 65
isMouseOver property 65
isMouseUp property 65
isThumbnail property 24
isPlaying method 39
K
KeyEvent object
about 44
canvas property 44
canvasPixelHeight property 44
canvasPixelWidth property 44
characterCode property 44
crKeyKeyDown property 45
currentTool property 45
shiftKeyKeyDown property 45
KeyEventHandler method 46
KeyEventHandler object
about 46
KeyEventHandler method 46
onEvent method 46
L
leftButtonDown property 65
length property 16, 116
lengthUnits property 92
Light object
about 47
ATTENUATION_ABC property 47
ATTENUATION_NONE property 47
attenuationA property 47
attenuationB property 47
attenuationC property 47
attenuationType property 47
brightness property 47
color property 47
direction property 48
innerConeAngle property 48
innerRadius property 48
outerConeAngle property 48
outerRadius property 48
position property 48
positionLocal property 48
type property 48
TYPE_INFINITE property 48
TYPE_POINT property 48
TYPE_SPOT property 48
LIGHT_MODE_BLUE property 92
LIGHT_MODE_BRIGHT property 92
LIGHT_MODE_CAD property 93
LIGHT_MODE_CUBE property 93
LIGHT_MODE_DAY property 92
LIGHT_MODE_FILE property 92
LIGHT_MODE_HEADLAMP property 93
LIGHT_MODE_NIGHT property 92
LIGHT_MODE_NONE property 92
LIGHT_MODE_RED property 92
LIGHT_MODE_RGB property 92
LIGHT_MODE_WHITE property 92
lights property 93
lightScaleFactor property 93
lightScheme property 93
loop property 36

M
Material object
about 49
ambientColor property 49
ambientTexture property 49
bumpTexture property 49
diffuseColor property 49
diffuseTexture property 49
emissiveColor property 49
emissiveTexture property 49
opacity property 49
opacityTexture property 49
phongExponent property 49
reflectionStrength property 49
reflectionTexture property 50
specularColor property 50
specularStrength property 50
material property 42, 64
materials property 93
Matrix4x4 method 51
Matrix4x4 object
about 51
determinant property 51
inverse property 51
invertInPlace method 52
isEqual method 52
Matrix4x4 method 51
multiply method 52
multiplyInPlace method 52
rotateAboutLine method 53
rotateAboutLineInPlace method 54
rotateAboutVector method 55
rotateAboutVectorInPlace method 55
rotateAboutX method 54
rotateAboutXInPlace method 55
rotateAboutY method 56
rotateAboutYInPlace method 56
rotateAboutZ method 56
rotateAboutZInPlace method 57
rotateWithQuaternion method 53
rotateWithQuaternionInPlace method 53
scale method 57
scaleComponent property 51
scaleInPlace method 58
set method 58, 59
setIdentity method 59
setView method 59
transformDirection method 60
transformPosition method 60
translate method 60
translateInPlace method 61
translation property 51
transpose property 51
transposeInPlace method 61
max property 20
MENU_ITEM_TYPE_CHECKED property 80
MENU_ITEM_TYPE_DEFAULT property 80
MenuEvent object
about 62
canvas property 62
currentTool property 62
menuItemChecked property 62
menuItemName property 62
MenuEventHandler method 63
MenuEventHandler object
about 63
MenuEventHandler method 63
onEvent method 63
menuItemChecked property 62
menuItemName property 62
Mesh object
about 64
computeBoundingBox method 64
material property 64
renderMode property 64
meshes property 93
metadataString property 69
methods
corrections in Acrobat 9.0 125
new in Acrobat 9.0 124
undocumented in earlier versions 125
min property 20
modulate property 110
MouseEvent object
about 65
canvas property 65, 104
canvasPixelHeight property 65, 104
canvasPixelWidth property 65, 104
ctrlKeyDown property 65, 104
currentTool property 65, 104
deltaY property 104
hits property 65
isDoubleClick property 65
isMouseDown property 65
isMouseHit property 65
isMouseMove property 65
isMouseOut property 65
isMouseOver property 65
isMouseUp property 65
leftButtonDown property 65
mouseX property 65
mouseY property 65
rightButtonDown property 66
shiftKeyDown property 66, 104

MouseEventHandler object
about 67
MouseEventHandler method 68
onEvent method 68
onMouseDoubleClick property 67
onMouseDown property 67
onMouseHit property 67
onMouseMove property 67
onMouseOut property 67
onMouseOver property 67
onMouseUp property 67
reportAllTargets property 68

Node object
about 69
detachFromCurrentAnimation method 70
directionLocal property 48
firstChild property 69
hitEnabled property 69
info property 69
metadataString property 69
nextSibling property 69
opacity property 69
parent property 69
transform property 69
visible property 70
wireframeColor property 70

N
name property 100
near property 21, 24
new methods
Acrobat 9.0 124
new objects
Acrobat 8.1 125
Acrobat 9.0 124
new properties
Acrobat 8.0 126
Acrobat 8.1 125
Acrobat 9.0 124
nextSibling property 69

O
objectGUID property 100
objectID property 100
objects
deprecated in Acrobat 8.1 126
new in Acrobat 8.1 125
new in Acrobat 9.0 124
onEvent method 25, 34, 46, 63, 68, 76, 105, 107, 109, 113, 115
onMouseDoubleClick property 67
onMouseDown property 67
onMouseHit property 67
onMouseMove property 67
onMouseOut property 67
onMouseOver property 67
onMouseUp property 67
opacity property 36, 69
opacityTexture property 49
outerConeAngle property 48
outerRadius property 48
outlineAngle property 94
overrideFlyTool property 80
overrideNavTools property 80
overridePanTool property 81
overrideRotateTool property 81
overrideScrollWheel property 81
overrideSelection property 81
overrideSpinTool property 81
overrideViewChange property 81
overrideWalkTool property 81
overrideZoomTool property 81

P
pan method 39
parent property 69
pause method 86
percentLoaded property 36
phongExponent property 49
play method 39, 86
playing property 36
pointsRenderModeColor property 77
position property 21, 42, 48
positionLocal property 21, 48
print method 31
println method 31
Procedural object 71
projectionType property 22, 24
properties
  corrections in Acrobat 9.0 125
  deprecated in Acrobat 8.1 126
  new in Acrobat 8.0 126
  new in Acrobat 8.1 125
  new in Acrobat 9.0 124
  undocumented in earlier versions 125

Q
  quality property 36
  Quaternion method 72
  Quaternion object
    about 72
    interpolate method 73
    interpolateInPlace method 73
    normalize method 73
    Quaternion method 72

R
  readyState property 36
  reflectionStrength property 49
  reflectionTexture property 50
  refresh method 87
  remove method 27
  removeAll method 102
  removeByIndex method 102
  removeCustomMenuItem method 87
  removeCustomToolButton method 88
  removeEventHandler method 87
  removeItem method 103
  RENDER_MODE_BOUNDING_BOX property 95
  RENDER_MODE_DEFAULT property 95
  RENDER_MODE_HIDDEN_WIREFRAME property 96
  RENDER_MODE_ILLUSTRATION property 96
  RENDER_MODE_SHADED_ILLUSTRATION property 96
  RENDER_MODE_SHADED_VERTICES property 95
  RENDER_MODE_SHADED_WIREFRAME property 95
  RENDER_MODE_SOLID property 95
  RENDER_MODE_SOLID_WIREFRAME property 96
  RENDER_MODE_SOLID_OUTLINE property 96
  RENDER_MODE_TRANSPARENT property 96
  RENDER_MODE_TRANSPARENT_BOUNDING_BOX property 95
  RENDER_MODE_TRANSPARENT_BOUNDING_BOX_OUTLINE property 95
  RENDER_MODE_TRANSPARENT_WIREFRAME property 96
  RENDER_MODE_VERTICES property 95
  RENDER_MODE_WIREFRAME property 95
  renderDoubleSided property 94
  RenderEvent object
    about 75
    canvas property 75
    canvasPixelHeight property 75
    canvasPixelWidth property 75
    currentTool property 75
    RenderEvent method 76
    RenderEventHandler method 76
    RenderEventHandler object
      about 77
      renderMode property 64, 94
      RenderOptions object
        about 77
        boundingBoxColor property 77
        clippingPlaneColor property 77
        clippingPlaneIntersectionColor property 77
        defaultAmbientColor property 77
        defaultDiffuseColor property 77
        defaultEmissiveColor property 77
        defaultSpecularColor property 77
        illustrationRenderModeFaceColor property 77
        illustrationRenderModeLineColor property 77
        pointsRenderModeColor property 77
        shadedIllustrationRenderModeLineColor property 77
        solidGridColorEven property 78
        solidGridColorOdd property 78
        solidRenderModeLineColor property 78
        transparentBoundsRenderModeColor property 78
        transparentGridColorEven property 78
        transparentGridColorOdd property 78
        wireframeRenderModeColor property 78
        xAxisColor property 78
        yAxisColor property 78
        zAxisColor property 78
      reportAllTargets property 68
      RESOLUTION_TYPE_CUSTOM property 36
      RESOLUTION_TYPE_MOVIE property 36
      RESOLUTION_TYPE_WINDOW property 36
      resolutionType property 36
      Resource method 79
      Resource object
        about 79
        Resource method 79
        type property 79
        TYPE_IMAGE property 79
        TYPE_MODEL property 79
        TYPE_UNKNOWN property 79
      rewind method 39
      rightButtonDown property 66
      roll property 22
      rotateAboutLine method 53
      rotateAboutLineInPlace method 54
      rotateAboutVector method 55
      rotateAboutVectorInPlace method 55
      rotateAboutX method 54
      rotateAboutXInPlace method 55
      rotateAboutY method 56
      rotateAboutYInPlace method 56
      rotateAboutZ method 56
      rotateAboutZInPlace method 57
      rotateWithQuaternion method 53
      rotateWithQuaternionInPlace method 53
      Runtime object
        about 80
        addCustomMenuItem method 83
        addCustomToolButton method 83
        addEventHandler method 84
        BUTTON_TYPE_PUSH property 80
        BUTTON_TYPE_TOOL property 80
canvasCount property 80
ctrlKeyDown property 80
disableTool method 84
enableTool method 84
eventHandlerCount property 80
getEventHandler method 85
getRendererName method 85
instances property 80
MENU_ITEM_TYPE_CHECKED property 80
MENU_ITEM_TYPE_DEFAULT property 80
overrideFlyTools property 80
overrideNavTools property 80
overridePanTool property 81
overrideRotateTool property 81
overrideScrollWheel property 81
overrideSelection property 81
overrideSpinTool property 81
overrideViewChange property 81
overrideWalkTool property 81
overrideZoomTool property 81
pause method 86
play method 86
refresh method 87
removeCustomMenuItem method 87
removeCustomToolButton method 88
removeEventHandler method 87
setCustomMenuItemChecked method 88
shiftKeyDown property 81
speedThreshold property 82
strafeSpeed property 82
TOOL_NAME_MEASURE property 82
TOOL_NAME_PAN property 82
TOOL_NAME_ROTATE property 82
TOOL_NAME_SPIN property 82
TOOL_NAME_WALK property 82
TOOL_NAME_ZOOM property 82
version property 82
walkSpeed property 83
createLight method 98
createSquareMesh method 98
defaultRenderOptions property 91
GRID_MODE_OFF property 91
GRID_MODE_SOLID property 91
GRID_MODE_TRANSPARENT property 91
GRID_MODE_WIRE property 92
gridMode property 91
gridSize property 92
lengthUnits property 92
LIGHT_MODE_BLUE property 92
LIGHT_MODE_BRIGHT property 92
LIGHT_MODE_CAD property 93
LIGHT_MODE_CUBE property 93
LIGHT_MODE_DAY property 92
LIGHT_MODE_FILE property 92
LIGHT_MODE_HEADLAMP property 93
LIGHT_MODE_NIGHT property 92
LIGHT_MODE_NONE property 92
LIGHT_MODE_RED property 92
LIGHT_MODE_RGB property 92
LIGHT_MODE_WHITE property 92
lights property 93
lightScaleFactor property 93
lightScheme property 93
materials property 93
meshes property 93
nodes property 94
outlineAngle property 94
RENDER_MODE_BOUNDING_BOX property 95
RENDER_MODE_DEFAULT property 95
RENDER_MODE_HIDDEN_WIREFRAME property 96
RENDER_MODE_ILLUSTRATION property 96
RENDER_MODE_SHADED_ILLUSTRATION property 96
RENDER_MODE_SHADED_VERTICES property 95
RENDER_MODE_SHADED_WIREFRAME property 95
RENDER_MODE_SOLID property 95
RENDER_MODE_SOLID_WIREFRAME property 96
RENDER_MODE_SOLID_OUTLINE property 96
RENDER_MODE_TRANSPARENT property 96
RENDER_MODE_TRANSPARENT_BOUNDING_BOX property 95
RENDER_MODE_TRANSPARENT_BOUNDING_BOX_OUTLINE property 95
RENDER_MODE_TRANSPARENT_WIREFRAME property 96
RENDER_MODE_VERTICES property 95
RENDER_MODE_WIREFRAME property 95
renderDoubleSided property 94
smoothing property 97
smoothAngle property 97
smoothingOverride property 97
update method 99
SceneObject object
about 100
name property 100
Scene object
about 91
activateAnimation method 97
addFlashForeground method 97
addModel method 98
ambientIlluminationColor property 91
animations property 91
cameras property 91
computeBoundingBox method 99
createClippingPlane method 98
createLight method 98
createSquareMesh method 98
defaultRenderOptions property 91
GRID_MODE_OFF property 91
GRID_MODE_SOLID property 91
GRID_MODE_TRANSPARENT property 91
GRID_MODE_WIRE property 92
gridMode property 91
gridSize property 92
lengthUnits property 92
LIGHT_MODE_BLUE property 92
LIGHT_MODE_BRIGHT property 92
LIGHT_MODE_CAD property 93
LIGHT_MODE_CUBE property 93
LIGHT_MODE_DAY property 92
LIGHT_MODE_FILE property 92
LIGHT_MODE_HEADLAMP property 93
LIGHT_MODE_NIGHT property 92
LIGHT_MODE_NONE property 92
LIGHT_MODE_RED property 92
LIGHT_MODE_RGB property 92
LIGHT_MODE_WHITE property 92
lights property 93
lightScaleFactor property 93
lightScheme property 93
materials property 93
meshes property 93
nodes property 94
outlineAngle property 94
RENDER_MODE_BOUNDING_BOX property 95
RENDER_MODE_DEFAULT property 95
RENDER_MODE_HIDDEN_WIREFRAME property 96
RENDER_MODE_ILLUSTRATION property 96
RENDER_MODE_SHADED_ILLUSTRATION property 96
RENDER_MODE_SHADED_VERTICES property 95
RENDER_MODE_SHADED_WIREFRAME property 95
RENDER_MODE_SOLID property 95
RENDER_MODE_SOLID_WIREFRAME property 96
RENDER_MODE_SOLID_OUTLINE property 96
RENDER_MODE_TRANSPARENT property 96
RENDER_MODE_TRANSPARENT_BOUNDING_BOX property 95
RENDER_MODE_TRANSPARENT_BOUNDING_BOX_OUTLINE property 95
RENDER_MODE_TRANSPARENT_WIREFRAME property 96
RENDER_MODE_VERTICES property 95
RENDER_MODE_WIREFRAME property 95
renderDoubleSided property 94
smoothing property 97
smoothAngle property 97
smoothingOverride property 97
update method 99
SceneObject object
about 100
name property 100
Scene object
about 91
activateAnimation method 97
addFlashForeground method 97
addModel method 98
ambientIlluminationColor property 91
animations property 91
cameras property 91
computeBoundingBox method 99
createClippingPlane method 98
createLight method 98
createSquareMesh method 98
defaultRenderOptions property 91
GRID_MODE_OFF property 91
GRID_MODE_SOLID property 91
GRID_MODE_TRANSPARENT property 91
GRID_MODE_WIRE property 92
gridMode property 91
gridSize property 92
lengthUnits property 92
LIGHT_MODE_BLUE property 92
LIGHT_MODE_BRIGHT property 92
LIGHT_MODE_CAD property 93
LIGHT_MODE_CUBE property 93
LIGHT_MODE_DAY property 92
LIGHT_MODE_FILE property 92
LIGHT_MODE_HEADLAMP property 93
LIGHT_MODE_NIGHT property 92
LIGHT_MODE_NONE property 92
LIGHT_MODE_RED property 92
LIGHT_MODE_RGB property 92
LIGHT_MODE_WHITE property 92
lights property 93
lightScaleFactor property 93
lightScheme property 93
materials property 93
meshes property 93
nodes property 94
outlineAngle property 94
RENDER_MODE_BOUNDING_BOX property 95
RENDER_MODE_DEFAULT property 95
RENDER_MODE_HIDDEN_WIREFRAME property 96
RENDER_MODE_ILLUSTRATION property 96
RENDER_MODE_SHADED_ILLUSTRATION property 96
RENDER_MODE_SHADED_VERTICES property 95
RENDER_MODE_SHADED_WIREFRAME property 95
RENDER_MODE_SOLID property 95
RENDER_MODE_SOLID_WIREFRAME property 96
RENDER_MODE_SOLID_OUTLINE property 96
RENDER_MODE_TRANSPARENT property 96
RENDER_MODE_TRANSPARENT_BOUNDING_BOX property 95
RENDER_MODE_TRANSPARENT_BOUNDING_BOX_OUTLINE property 95
RENDER_MODE_TRANSPARENT_WIREFRAME property 96
RENDER_MODE_VERTICES property 95
RENDER_MODE_WIREFRAME property 95
renderDoubleSided property 94
smoothing property 97
smoothAngle property 97
smoothingOverride property 97
update method 99
SceneObject object
about 100
name property 100
objectGUID property 100
objectID property 100
SceneObjectList object 101
about 101
count property 101
getByGUID method 101
getById method 101
getByIndex method 101
getName method 102
removeAll method 102
removeByIndex method 102
removeItem method 103
SceneObjectList object 101
about 101
count property 101
getByGUID method 101
getById method 101
getByIndex method 101
getName method 102
removeAll method 102
removeByIndex method 102
removeItem method 103
ScrollWheelEvent object
about 104
canvas property 104
canvasPixeleight property 104
canvasPixelWidth property 104
ctrlKeyDown property 104
currentTool property 104
deltaY property 104
shiftKeyDown property 104
ScrollWheelEventHandler object
about 105
onEvent method 105
ScrollWheelEventHandler method 105
scrollWheelSpeed property 81
selected property 106
selectedNode property 96
SelectionEvent object
about 106
node property 106
selected property 106
SelectionEventHandler method 107
SelectionEventHandler object
about 107
onEvent method 107
SelectionEventHandler method 107
set method 28, 58, 59, 120, 121
set3 method 29, 121
setCamera method 26
setColor method 17
target property 22, 33, 34, 42, 68
targetDistance property 24
targetPosition property 22
targetPositionLocal property 22
Texture object
about 110
amount property 110
angle property 110
clampU property 110
clampV property 110
getimage method 110
image property 110
modulate property 110
setimage method 111
uOffset property 110
uScale property 110
use3DSStyleMapping property 110
vOffset property 110
vScale property 110
texCoordinate property 42
time event
measuring using the JavaScriptDate object 14
time property 112
TimeEvent object
about 112
deltaTime property 112
time property 112
TimeEventHandler method 113
TimeEventHandler object
about 113
onEvent method 113
TimeEventHandler method 113
TOOL_NAME_FLY property 82
TOOL_NAME_MEASURE property 82
TOOL_NAME_PAN property 82
TOOL_NAME_ROTATE property 82
TOOL_NAME_SPIN property 82
TOOL_NAME_WALK property 82
TOOL_NAME_ZOOM property 82
ToolEvent object
  about 114
  canvas property 114
  canvasPixelHeight property 114
  canvasPixelWidth property 114
  currentTool property 114
  toolName property 114
ToolEventHandler method 115
ToolEventHandler object
  about 115
  onEvent method 115
  ToolEventHandler method 115
toolName property 114
totalFrames property 37
transform property 24, 69
transformDirection method 60
transformPosition method 60
translate method 60
translateInPlace method 61
translation property 51
transparentBoundsRenderModeColor property 78
transparentGridColorEven property 78
transparentGridColorOdd property 78
transpose property 51
transposeInPlace method 61
type property 33, 48, 79, 108
TYPE_COMMAND property 33
TYPE_IMAGE property 79
TYPE_INFINITE property 48
TYPE_LOAD property 108
TYPE_MODEL property 79
TYPE_ORTHOGRAPHIC property 22
TYPE_PERSPECTIVE property 22
TYPE_POINT property 48
TYPE_PROGRESS property 33
TYPE_SAVE property 108
TYPE_SPOT property 48
TYPE_STATECHANGE property 33
TYPE_UNKNOWN property 79

V
defire property 33
Vector3 method 116
Vector3 object
  about 116
  add method 117
  addInPlace method 117
  addScaled method 117
  addScaledInPlace method 118
  blend method 118
  blendInPlace method 118
cross method 119
dot method 119
  length property 116
  normalize method 119
  scale method 120
  scaleInPlace method 120
  set method 120, 121
  set3 method 121
  subtract method 122
  subtractInPlace method 122
  Vector3 method 116
  x property 116
  y property 116
  z property 116
version property 82
viewPlaneSize property 22, 24
visible property 70
vOffset property 110
vScale property 110

W
walkSpeed property 83
width property 43
wireframeColor property 70
wireframeRenderModeColor property 78

X
x property 37, 116
xAxisColor property 78

Y
y property 37, 116
yAxisColor property 78

Z
z property 116
zAxisColor property 78
zoom method 41