Flash ActionScript Quick Reference

Author: Jialong He
Jialong_he@yahoo.com
http://tiger.la.asu.edu

Introduction

Using Flash to create animations on the web is popular because the flash player is installed on most computers and the published flash file (SWF file) is small. Flash has a powerful scripting language called ActionScript. You can use write script to manipulate and control objects on the stage. Its syntax is similar to JavaScript (or C++).

Note: this quick reference is based on ActionScript 2.0 language reference.

Script Example

A script can be associated with a keyframe or with an object. To test the following script, copy it to the action panel. Press "Ctrl + Enter" to start the flash file.

```
//====================
//  A simple ActionScript
//====================
for (Cnt=1; Cnt<10; Cnt++) {
    trace(Math.random());
}
trace ("Hello, World!");
```

Operator

+, -, *, /, %	Add, Subtract, Multiply, Division, Remainder
+=, -=, *=, =/, =%	Combine with assignment operator
++, --, []	Increase, decrease, Array access
==, !=, <, <=, >, >=	Comparison, equal, not equal, less than, …
!, &&, ||	Logical NOT, AND, OR
<<, >>, >>>	Bit shift, left, right, right unsigned
~, &, |, ^	Bitwise NOT, AND, OR, XOR
new, delete	Allocate (delete) an object
typeof, instanceof	get expression type, test an instance
//, /* */	One line and multiple line comments

Constants and Compiler Directives

true, false, undefined, null, NaN, Infinity newline	Predefined constants
#initclip statements(s) #endinitclip	Initialization actions are executed only once when a SWF file is played
#include "filename.as"	Include external ActionScript

Program Flow Control

if (condition){ statement(s); } else { statement(s); }

```
if (age>=18) {
  trace("welcome, user");
}
else {
  trace("sorry, junior");
}
```

for (init; condition; next) { statement(s); }

```
for (Cnt = 1; Cnt<10; Cnt++) {
  trace(Cnt);
}
```

switch (expression){ caseClause: [defaultClause:] }

```
Switch (myChar) {
  case "A" :
  case "a" :
    trace("you pressed A or a");
    break;
  default :
    trace("you did not press A");
}
```

for (var in object) { statement(s); }

```
var myObject:Object = {Name:"Tara",
age:27, city:"San Francisco"};

for (var prop in myObject) {
  trace(myObject[prop]);
}
```

while(condition) { statement(s); }

```
var Cnt:Number = 0;
while (Cnt < 20) {
  trace(Cnt);
  i += 3;
}
```

do { *statement(s)* } while (*condition*)

```
var myVar:Number = 0;
do {
  trace(myVar);
  myVar++;
} while (myVar < 5);
```

function FName(P){ statement(s) }

```
function mySquared(x:Number) {
  return Math.pow(x, 2);
}
```

class, interface, implement, dynamic, extend, private, public, intrinsic — Define custom class related statements

Global Functions

play, stop, nextFrame, prevFrame gotoAndPlay, gotoAndStop nextScene, prevScene	Main timeline movie clip play head control
loadMovie, loadMovieNum unloadMovie, unloadMovieNum	Loads (unload) a SWF, JPEG, GIF, or PNG file from local disk or web server into a movie clip
loadVariables, loadVariablesNum	Reads data from an external file either on local disk or on web server

setInterval, clearInterval
Repeatly execute a function (or an object).

```
\\========= Example=========
var intervalId:Number;
var count:Number = 0;
var maxCount:Number = 10;
var duration:Number = 20;

function myCallback():Void {
  trace(count);
    if(count >= maxCount) {clearInterval(intervalId);}
  count++;
}
intervalId = setInterval(this, "myCallback", duration);
```

escape, unescape	Converts the parameter to a string and encodes it in a URL-encoded format, where all nonalphanumeric characters are replaced with % hexadecimal sequences (e.g, @ to %40).
getProperty, setProperty	Get (set) movie clip property
getURL	Load a web page in browser
on (event) { }	Mouse/ket event handler Press, release, releaseOutside, rollOut, rollOver, dragOut, dragOver, keyPress e.g.; on (press) { startDrag(this); }

onClipEvent(movieEvent: Object) { statements; }	Movie clip event handler. load, unload, enterFrame, mouseMove, MouseDown, MouseUp, KeyDown, KeyUp, Data.
	
	onClipEvent (keyDown) {
	  if (Key.getCode() == Key.RIGHT) {
	  this._parent.nextFrame();
	  } else if (Key.getCode() == Key.LEFT) {
	  this._parent.prevFrame();
	  }
	}

startDrag, stopDrag	Makes the target movie clip draggable while the movie plays.
	
	my_mc.onPress = function () {
	startDrag(this);
	}
	my_mc.onRelease = function() {
	stopDrag();
	}

fscommand	Lets the SWF file communicate with either Flash Player or the program that is hosting Flash Player, such as a web browser. e.g, fscommand("fullscreen", true);
isFinite, isNaN	Test number

getVersion , targetPath, trace, getTimer, removeMovieClip duplicateMovieClip	Miscellaneous functions.

Global Properties

_global	A reference to the global object that holds the core ActionScript classes, such as String, Object, Math, and Array.
_parent	Specifies or returns a reference to the movie clip or object that contains the current movie clip or object.
_root	Specifies or returns a reference to the root movie clip Timeline.
This	References an object or movie clip instance.

Common Classes

Object

constructor	Object
Properties	constructor, _proto_, prototype, _resolve,
Methods	addProperty, hasOwnProperty, isPropertyEnumerable, isPrototypeOf, registerClass, toString, unwatch, valueOf, watch

Array

Properties	CASEINSENSITIVE, DESCENDING, length, NUMRIC, RETURNINDEXEDARRAY, UNIQUESORT
Methods	concat, join, pop, push, reverse, shift, slice, sort, sortOn, splice, toString, unshift.

Example	var myA:Array = new Array("a","b","c");
	var myN:Array = new Array(1,2,3);
	var myAN:Array =myA.concat(myN);
	trace(myAN.length);
	// Creates array [a,b,c,1,2,3].

Date

Properties	Only have properties inherited from Object.
Methods	getDate, getDay, getFullYear, getHours, getMilliseconds, getMinutes, getMonth, getSeconds, getTime, getTimezoneOffset, getYear, setDate, setFullYear, setHours, setMilliseconds, setMinutes, setMonth, setSeconds, setTime, setYear, toString, valueOf, (most functions have UTC ones)

Example	var my_date:Date = new Date(2004,4,25);
	trace(my_date.getYear()); // output: 104
	trace(my_date.getFullYear()); // output: 2004
	my_date.setYear(99);
	trace(my_date.getYear()); // output: 99
	trace(my_date.getFullYear()); // output: 1999

Math

Properties	E, LN10, LN2, LOG10E, LOG2E, PI, SORT1_2, SORT2
Methods	abs, acos, asin, atan, atan2, ceil, cos, exp, floor, log, max, min, pow, random, round, sin, sqrt, tan

Example	trace(Math.log(0)); // output: -Infinity
	trace(Math.atan(-1)); // output: -0.785398163397448

String

Properties	length
Methods	charAt, charCodeAt, concat, fromCharCode, indexOf, lastIndexOf, slice, split, substr, substring, toLowerCase, toString, toUpperCase, valueOf

Example	var my_str:String = new String("Hello world");
	var mySubstring:String = new String();
	
	mySubstring = my_str.substr(6,5);
	trace(mySubstring); // output: world
	
	trace (mySubstring.toUpperCase()); //WORLD

Stage

Properties	align, height, scaleMode, showMenu, width
Event	onResize
Methods	addListener, removeListener

Example	Stage.scaleMode = "noScale"
	var myListener:Object = new Object();
	myListener.onResize = function () {
	trace("Stage size is now " + Stage.width + " by " + Stage.height);
	}
	Stage.addListener(myListener);

Key

Properties	BACKSPACE, CAPSLOCK, CONTROL, DELETEKEY, DOWN, END, ENTER, ESCAPE, HOME, INSERT, LEFT, PGDN, PGUP, RIGHT, SHIFT, SPACE, TAB, UP, _listeners
Events	onKeyDown, onKeyUp
Methods	addListener, getAscii, getCode, isAccessible, isDown, isToggled, removeListener

Example	var myListener:Object = new Object();
	myListener.onKeyDown = function () {
	trace ("You pressed a key.");
	}
	myListener.onKeyUp = function () {
	trace ("You released a key.");
	}
	Key.addListener(myListener);

Mouse

Events	onMouseDown, onMouseMove, onMouseUp, onMouseWheel
Methods	addListener, hide, removeListener, show

Example	var mouseListener:Object = new Object();
	
	mouseListener.onMouseDown = function() {
	trace("Mouse down");
	};
	mouseListener.onMouseMove = function() {
	trace(_xmouse);
	trace(_ymouse);
	};
	mouseListener.onMouseUp = function() {
	trace("Mouse up");
	};
	Mouse.addListener(mouseListener);

Button

Properties	_alpha, blendMode, cacheAsBitmap, enabled, filters, _focusrect, _height, _highquality, menu, _name, _parent, _quality, _rotation, scale9Grid, _soundbuftime, tabEnabled, tabIndex, _target, trackAsMenu, _url, useHandCursor, _visible, _width, _x, _xmouse, _xscale, _y, _ymouse, _yscale
Events	onDragOut, onDragOver, onKeyDown, onKeyUp, onKillFocus, onPress, onRelease, onReseaseOutside, onRollOut, onRollOver, onSetFocus
Methods	getDepth

Example	myBtn1_btn.enabled = true;
	myBtn2_btn.enabled = false;
	
	myBtn1_btn.onRelease = function() {
	trace("you clicked : " + this._name);
	};
	myBtn2_btn.onRelease = function() {
	trace("you clicked : " + this._name);
	};

TextFormat

Constructor	TextFormat
Properties	align, blockIndent, bold, bullet, color, font, indent, italic, kerning, leading, leftMargin, letterSpacing, rightMargin, size, tabStops, target, underline, url
Methods	getTextExtent

Example	var my_fmt:TextFormat = new TextFormat();
	my_fmt.bold = true; my_fmt.font = "Arial";
	my_fmt.size = 12; my_fmt.color = 0xFF0000;
	
	this.createTextField("stats_txt", 5000, 10, 0, 530, 22);
	stats_txt.setNewTextFormat(my_fmt);

TextField

Properties _alpha, antiAliasType, autoSize, background, backgroundColor, border, borderColor, bottomScroll, condeseWhite, enbedFonts, filter, gridFitType, _height, _highquality, hscroll, html, htmlText, length, maxChars, maxhscroll, maxscroll, menu, mouseWheelEnabled, multiline, _name, _parent, password, _quality, restrict, _rotation, scroll, selectable, sharpness, _soundbuftime, styleSheet, tabEnabled, tabIndex, _target, text, textColor, textHeight, textWidth, thickness, type, _url, variable, _visible, _width, wordWrap, _x, _xmouse, _xscale, _y, _ymouse, _yscale

Event onChanged, onKillFocus, onScroller, onSetFocus

Methods addListener, getDepth, getFontList, getNewTextFormat, getTextFormat, removeLisener, removeTextField, replaceSel, replaceText, setNewTextFormat, setTextFormat

Example
```
my_txt.border = true;
my_txt.type = "input";

my_txt.onChanged = function(textfield_txt:TextField) {
   trace(textfield_txt._name+" changed");
};

var txtListener:Object = new Object();
txtListener.onChanged = function(textfield_txt:TextField) {
   trace(textfield_txt._name+" changed and notified
myListener");
};
my_txt.addListener(txtListener);
```

Sound

constructor Sound

Properties duration, id3, position

Events onID3, onLoad, onSoundComplete

Methods attachSound, getBytesLoaded, getBytesTotal, getPan, getTransform, getVolume, loadSound, setPan, setTransform, setVolume, start, stop

Example
```
var my_sound:Sound = new Sound();
my_sound.attachSound("logoff_id");

my_sound.onSoundComplete = function() {
trace("mySoundID completed");
};

my_sound.start();
```

Video

Properties _alpha, deblocking, _height, height, _name, _parent, _rotation, smoothing, _visible, _width, width, _x, _xmouse, _xscale, _y, _ymouse, _yscale

Methods attachVideo, clear

Example
```
var my_video:Video;
var my_nc:NetConnection = new NetConnection();
my_nc.connect(null);
var my_ns:NetStream = new NetStream(my_nc);
my_video.attachVideo(my_ns);
my_ns.play("video1.flv");
```

MovieClip

Properties _alpha, blendMode, cacheAsBitmap, _currentframe, _droptarget, enabled, filters, focusEnabled, _focusrect, _framesloaded, _height, _highquality, hitArea, _lockroot, menu, _name, opaqueBackground, _parent, _quality, _rotation, scale9Grid, scrollRect, _soundbuftime, tabChildren, tabEnabled, tabIndex, _target, _totalframes, trackAsMenu, transform, _url, useHandCursor, _visible, _width, _x, _xmouse, _xscale, _y, _ymouse, _yscale

Methods attachAudio, attachBitmap, attachMovie, beginBitmapFill, beginFill, beginGradientFill, clear, createEmptyMovieClip, createTextField, curveTo, duplicateMovieClip, endFill, getBounds, getBytesLoaded, getBytesTotal, getDepth, getInstanceAtDepth, getNextHightsDepth, getRect, getSWFVersion, getTextSnapshot, getURL, globalToLocal, gotoAndPlay, gotoAndStop, hitTest, lineGradientStyle, lineStyle, lineTo, loadMoive, loadVariables, localToGlobal, moveTo, nextFrame, play, prevFrame, removeMovieClip, setMask, setMask, startDrag, stop, stopDrag, swapDepths, unloadMovie

Events onData, onDragOut, onDragOver, onEnterFrame, onKeyDown, onKeyUp, onKillFocus, onLoad, onMouseDown, onMouseMove, onMouseUp, onPress, onRelease, onReleaseOutside, onRollOut, onRollOver, onSetFocus, onUnload

Example
```
this.createEmptyMovieClip("triangle",
this.getNextHighestDepth());

triangle.beginFill(0x0000FF, 100);
triangle.moveTo(10, 10);
triangle.lineTo(10, 100);
triangle.lineTo(100, 10);
triangle.lineTo(10, 10);

triangle.onRollOver = function() {
   this._alpha = 50;
};
triangle.onRollOut = function() {
   this._alpha = 100;
};
```

XMLNode

Constructor XMLNode

Properties attributes, childNodes, firstChild, lastChild,localName, namespaceURI, nextSibling, nodeName, nodeType, nodeValue, parendNode, prefix, previousSibling

Methods appendChild, cloneNode, getnamespaceForPrefix, getPrefixForNamespace, hasChildNodes, removeNode, toString

XML

Constructor XML

Properties contenttype, docTypeDecl, idMap, ignoreWhite, loaded, status, xmlDecl

Events onData, onHTTPStatus, onLoad

Methods addRequestHeader, createElement, createTextNode, getBytesLoaded, getBytesTotal, load, parseXML, send, sendAndLoad

Example
```
var myXML:XML = new XML();
myXML.ignoreWhite = true;

myXML.onLoad = function () {
                trace(this.childNodes);
               }

myXML.load ("flute.xml");
```

XMLsocket

Constructor XMLSocket

Events onClose, onConnect, onData, onXML

Methods close, connect, send

Example
```
var socket:XMLSocket = new XMLSocket()
socket.onConnect = function (success:Boolean) {
   if (success) {
   trace ("Connection succeeded!")
   } else {
   trace ("Connection failed!")
   }
}
if (!socket.connect(null, 2000)) {
   trace ("Connection failed!")
}
```