

Autodesk® Scaleform®

Scaleform 4.2 AS3 익스텐션 참고사항

Scaleform 4.2 에서 사용하는 ActionScript 3.0 익스텐션을 설명한 문서입니다.

작성자: Artem Bolgar, Prasad Silva

버전: 1.04

최종 수정일: 2012 년 9 월 17 일

저작권 정보

Autodesk® Scaleform® 4.2

© 2012 Autodesk, Inc. All rights reserved. Except as otherwise permitted by Autodesk, Inc., this publication, or parts thereof, may not be reproduced in any form, by any method, for any purpose.

Certain materials included in this publication are reprinted with the permission of the copyright holder.

The following are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and other countries: 123D, 3ds Max, Algor, Alias, AliasStudio, ATC, AUGI, AutoCAD, AutoCAD Learning Assistance, AutoCAD LT, AutoCAD Simulator, AutoCAD SQL Extension, AutoCAD SQL Interface, Autodesk, Autodesk Homestyler, Autodesk Intent, Autodesk Inventor, Autodesk MapGuide, Autodesk Streamline, AutoLISP, AutoSketch, AutoSnap, AutoTrack, Backburner, Backdraft, Beast, Beast (design/logo) Built with ObjectARX (design/logo), Burn, Buzzsaw, CAiCE, CFdesign, Civil 3D, Cleaner, Cleaner Central, ClearScale, Colour Warper, Combustion, Communication Specification, Constructware, Content Explorer, Creative Bridge, Dancing Baby (image), DesignCenter, Design Doctor, Designer's Toolkit, DesignKids, DesignProf, DesignServer, DesignStudio, Design Web Format, Discreet, DWF, DWG, DWG (design/logo), DWG Extreme, DWG TrueConvert, DWG TrueView, DWFx, DXF, Ecotect, Evolver, Exposure, Extending the Design Team, Face Robot, FBX, Fempro, Fire, Flame, Flare, Flint, FMDesktop, Freewheel, GDX Driver, Green Building Studio, Heads-up Design, Heidi, Homestyler, HumanIK, i-drop, ImageModeler, iMOUT, Incinerator, Inferno, Instructables, Instructables (stylized robot design/logo), Inventor, Inventor LT, Kynapse, Kynogon, LandXplorer, Lustre, MatchMover, Maya, Mechanical Desktop, MIMI, Moldflow, Moldflow Plastics Advisers, Moldflow Plastics Insight, Moondust, MotionBuilder, Movimento, MPA, MPA (design/logo), MPI (design/logo), MPX, MPX (design/logo), Mudbox, Multi-Master Editing, Navisworks, ObjectARX, ObjectDBX, Opticore, Pipeplus, Pixlr, Pixlr-o-matic, PolarSnap, Powered with Autodesk Technology, Productstream, ProMaterials, RasterDWG, RealDWG, Real-time Roto, Recognize, Render Queue, Retimer, Reveal, Revit, RiverCAD, Robot, Scaleform, Scaleform GFx, Showcase, Show Me, ShowMotion, SketchBook, Smoke, Softimage, Sparks, SteeringWheels, Stitcher, Stone, StormNET, Tinkerbox, ToolClip, Topobase, Toxik, TrustedDWG, T-Splines, U-Vis, ViewCube, Visual, Visual LISP, Vtour, WaterNetworks, Wire, Wiretap, WiretapCentral, XSI.

All other brand names, product names or trademarks belong to their respective holders.

Disclaimer

THIS PUBLICATION AND THE INFORMATION CONTAINED HEREIN IS MADE AVAILABLE BY AUTODESK, INC. "AS IS." AUTODESK, INC. DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE REGARDING THESE MATERIALS.

Autodesk Scaleform 연락처

문서	AS3 익스텐션 참고사항
주소	Autodesk Scaleform Corporation 6305 Ivy Lane, Suite 310 Greenbelt, MD 20770, USA
홈페이지	www.scaleform.com
전자 우편	info@scaleform.com
직통 전화	(301) 446-3200
팩스	(301) 446-3199

목차

1	소개	1
2	익스텐션	3
3	DisplayObjectEx 익스텐션	8
4	FocusEventEx 익스텐션	10
5	FocusManager 익스텐션.....	11
6	GamePad 익스텐션	19
7	GamePadAnalogEvent 익스텐션.....	20
8	InteractiveObjectEx 익스텐션.....	23
9	KeyboardEventEx 익스텐션	26
10	MouseEventEx 익스텐션	28
11	MouseEventEx 익스텐션.....	30
12	TextEventEx 익스텐션.....	33
13	시스템 익스텐션	35
14	TextFieldEx 익스텐션	36

1 소개

Autodesk Scaleform® SDK™는 클린룸 구현법에 기반을 둔 콘솔/PC 게임 개발용 경량 고성능 리치 미디어 Adobe® Flash® 벡터 그래픽 엔진입니다. Autodesk Scaleform 은 Adobe Creative Suite®와 같이 널리 인정받는 시각적 저작 도구의 사용 편의성, 확장성과 최첨단 게임 개발에 필요한 최신 하드웨어 그래픽 가속 기능을 결합한 제품입니다.

Flash 는 게임이나 응용 프로그램 개발이 아니라 웹 개발을 염두에 두고 만든 제품이므로 focus 처리, 마우스 지원, 텍스트 필드 지원 및 IME 입력 처리 등의 특정 영역에서 충분한 기능을 지원하지 않습니다. Autodesk Scaleform 은 ActionScript 클래스에 "익스텐션"을 추가하여 이들 영역의 기본 기능을 보완합니다. Scaleform Player 에서 익스텐션을 사용하려면 `scaleform.gfx.Extensions.enabled` 속성을 true 로 설정해야 합니다.

```
scaleform.gfx.Extensions.enabled = true;
```

또는

```
import scaleform.gfx.*;
Extensions.enabled = true;
```

ActionScript 3.0 컴파일러에서 익스텐션 클래스가 보이게 하려면 Flash 의 "소스 경로"에 `GFXSDK/Resources/AS3/CLIK` 경로를 추가해야 합니다. 경로를 추가하려면 Flash Studio 에서 'Edit' -> 'Preferences' -> 'ActionScript' -> 'ActionScript 3.0 Settings...' -> 'Source path'로 이동하십시오. 특별한 경우가 아니라면 익스텐션을 사용할 FLA 파일의 첫 프레임에 이 구문을 추가하는 것이 통례입니다. 이 속성을 설정하지 않으면 Scaleform Player 가 Flash 호환성을 최대한 높이기 위해 익스텐션과 관련된 모든 참조를 무시합니다.

이 문서에서 설명하는 익스텐션 기능은 Scaleform 에서만 구현되므로 표준 Flash Player 에서는 작동하지 않는다는 사실을 기억하십시오. 익스텐션마다 추가된 플레이어 SDK 의 릴리스 번호를 식별하는 Scaleform 버전 번호가 연결되었습니다. 이전 버전의 익스텐션은 Scaleform Player 에서 작동하지 않습니다.

Autodesk 는 릴리스가 거듭되더라도 익스텐션 API 를 유지하고 지원하도록 노력하겠지만, 향후 Scaleform API 를 변경하거나 삭제하거나 API 의 이름을 바꿀 권한이 있습니다. 중대한 변경 사항은 주요 릴리스에서 적용하도록 노력하고 이를 최대한 빨리 알려드리겠습니다.

2 익스텐션

enabled 정적 속성

```
enabled:Boolean [read-write]
```

Scaleform 버전: 4.0.12

익스텐션을 활성화하거나 비활성화합니다.

noInvisibleAdvance 정적 속성

```
noInvisibleAdvance:Boolean [read-write]
```

Scaleform 버전: 4.0.12

이 속성을 true 로 설정하면 화면에 보이지 않는 모든 Advance 연산을 하지 않습니다. 이 기능으로 숨겨진 동영상의 많이 포함된 SWF 의 성능을 개선할 수 있습니다. 참고로, Flash 는 동영상이 화면에 보이지 않더라도 Advance 연산을 계속 하기 때문에 타임라인 애니메이션이나 프레임의 AcitionScript 코드 등을 실행합니다. 그러므로 이 속성을 true 로 설정하면 Scaleform 과 Flash 에서 다르게 작동할 수 있습니다.

getTopMostEntity() 정적 메서드

```
public function getTopMostEntity(x : Number, y : Number, testAll: Boolean) :  
DisplayObject
```

Scaleform 버전: 4.0.13

(x, y) 좌표(스테이지 좌표 공간)의 제일 위에 있는 DisplayObject 인스턴스를 반환합니다. 이 메서드는 버튼 핸들러를 설정(예: CLICK, MOUSE_DOWN, ROLL_OVER 등)한 무비클립과 그렇지 않은 무비클립 사이에서 다르게 작동할 수 있습니다. 이렇게 구분하면 마우스 이벤트를 처리하는 핸들러가 없는 무비클립을 필터링하는 데 도움이 될 수 있습니다.

hitTest 는 x, y 좌표가 특정 객체 안에 있는지 확인하고 getTopMostEntity 는 지정된 x, y 좌표에 있는 실제 객체를 반환하므로 이 메서드는 MovieClip.hitTest 와 정반대로 작동합니다. 따라서

```
Extensions.getTopMostEntity(x, y).hitTest(x, y, true) == true
```

입니다.

매개변수

testAll : Boolean - false 라면 버튼 핸들러가 있는 무비클립만을, true 라면 모든 무비클립을 검색합니다. 이 매개변수를 지정하지 않으면 getTopMostEntity 에서 해당 매개변수의 값이 true 라고 가정합니다.

x : Number, y : Number - 무비클립을 검색할 스테이지 좌표입니다.

참조:

[getMouseTopMostEntity](#)

getMouseTopMostEntity() 정적 메서드

```
public function getTopMostEntity([testAll : Boolean],  
    [mouseIndex : uint]) : DisplayObject
```

Scaleform 버전: 4.0.13

마우스 커서 현재 위치의 제일 위에 있는 DisplayObject 인스턴스를 반환합니다. 이 함수는 getTopMostEntity 와 유사하지만 명시한 좌표 대신 마우스 좌표를 사용합니다.

매개변수

testAll : Boolean - false 라면 버튼 핸들러가 있는 무비클립만을, true 라면 모든 무비클립을 검색합니다. 이 매개변수를 지정하지 않으면 getTopMostEntity 에서 해당 매개변수의 값이 true 라고 가정합니다.

mouseIndex : Number - 0 을 기준으로 하는 마우스 인덱스입니다.

참조:

[getTopMostEntity](#)

getMouseCursorType() 정적 메서드

```
public function getMouseCursorType([mouseIndex : uint]) : String
```

Scaleform 버전: 4.0.13

지정한 마우스 컨트롤러의 현재 마우스 커서 유형을 반환하는 정적 메서드입니다. 이 메서드는 flash.ui.Mouse.cursor 속성과 비슷하지만, 여러 마우스의 커서를 변경할 수 있다는 차이점이 있습니다.

마우스 커서를 추적하고 선택적으로 마우스 커서가 바뀌지 않도록 하려면 MouseCursorEvent.CURSOR_CHANGE 익스텐션 이벤트를 사용합니다.

매개변수

mouseIndex : Number – 0 을 기준으로 하는 마우스 인덱스입니다. 기본값은 0 입니다.

반환 값

커서 유형을 반환합니다. 다음과 같은 flash.ui.MouseCursor 정적 상수를 참조하십시오.

- flash.ui.MouseCursor.ARROW : String = "arrow"
- flash.ui.MouseCursor.BUTTON : String = "button"
- flash.ui.MouseCursor.HAND : String = "hand"
- flash.ui.MouseCursor.IBEAM : String = "ibeam"

참조:

[setMouseCursorType](#)
[MouseCursorEvent](#)

setMouseCursorType() 정적 메서드

```
public function setMouseCursorType(cursor : String, [mouseIndex : uint]) : void
```

Scaleform 버전: 4.0.13

cursor 매개변수에 따라 마우스 커서를 바꾸는 정적 메서드입니다. 이 메서드는 flash.ui.Mouse.cursor 속성과 비슷하지만, 여러 마우스의 커서를 변경할 수 있다는 차이점이 있습니다.

마우스 커서를 추적하고 선택적으로 마우스 커서가 바뀌지 않도록 하려면
MouseEvent.CURSOR_CHANGE 익스텐션 이벤트를 사용합니다.

매개변수

cursor : uint - 커서 유형입니다. 다음과 같은 flash.ui.MouseCursor 정적 상수를 참조하십시오.

- flash.ui.MouseCursor.ARROW : String = "arrow"
- flash.ui.MouseCursor.BUTTON : String = "button"
- flash.ui.MouseCursor.HAND : String = "hand"
- flash.ui.MouseCursor.IBEAM : String = "ibeam"

mouseIndex : Number - 0 을 기준으로 하는 마우스 인덱스입니다. 기본값은 0 입니다.

참조:

[getMouseEventType](#)
[MouseEvent](#)

numControllers 정적 속성

numControllers:uint [read]

Scaleform 버전: 4.0.12

시스템이 검색한 컨트롤러의 개수를 반환합니다.

setEdgeAAMode() 정적 메서드

public function setEdgeAAMode(displayObj:DisplayObject, mode:uint):void

Scaleform 버전: 4.0.12

특정 디스플레이 객체와 그 자식 객체의 EdgeAA 모드를 설정합니다. 다음은 사용할 수 있는 모드 값입니다.

- scaleform.gfx.Extensions.EDGEAA_INHERIT = 0; 부모로부터 EdgeAA 모드를 상속합니다.
기본값은 on 입니다.
- scaleform.gfx.Extensions.EDGEAA_ON = 1; 대상 디스플레이 객체와 그 자식 객체의 EdgeAA 모드가 비활성화 되지 않았다면 이를 활성화합니다(EDGEAA_DISABLE 참조).

- `scaleform.gfx.Extensions.EDGEAA_OFF = 2`; 대상 디스플레이 객체와 그 자식 객체에 EdgeAA 를 사용하지 마십시오.
- `scaleform.gfx.Extensions.EDGEAA_DISABLE = 3`; 대상 디스플레이 객체와 그 자식 객체가 상속한 `EDGEAA_ON` 모드 값을 무시하고 EdgeAA 를 비활성화합니다.

매개변수

`dispObj : DisplayObject` – 새 EdgeAA 모드 값을 적용할 대상 디스플레이 객체입니다.

`mode : uint` – EdgeAA 모드 값입니다.

참조:

[getEdgeAAMode](#)

getEdgeAAMode() 정적 메서드

```
static public function getEdgeAAMode(dispObj:DisplayObject): uint
```

Scaleform 버전: 4.0.12

특정 디스플레이 객체에서 EdgeAA 모드 값을 가져옵니다.

매개변수

`dispObj : DisplayObject` – EdgeAA 모드 값을 가져올 대상 디스플레이 객체입니다.

참조:

[setEdgeAAMode](#)

3 DisplayObjectEx 익스텐션

setRendererString 정적 메서드

```
static public function setRendererString(o:DisplayObject, s:String)
```

Scaleform version: 4.0.17

이 속성은 모든 MovieClip 인스턴스에 대해 액션스크립트에서 렌더러로 커스텀 명령을 전송하도록 한다. 속성이 설정되면 문자열 값은 사용자 데이터 형태로 렌더러에 전송된다.

기본값은 없다.

예:

```
import scaleform.gfx.*;

// m is a MovieClip on stage.

DisplayObjectEx.setRendererString(m, "Abc");
trace(DisplayObjectEx.getRendererString(m));
```

setRendererFloat 정적 메서드

```
static public function setRendererFloat(o:DisplayObject, f:Number)
```

Scaleform version: 4.0.17

이 속성은 MovieClip 인스턴스에 대해 액션스크립트에서 커스텀 명령을 렌더러로 전송하도록 한다. 속성이 설정되면 Float 값은 사용자 데이터 형태로 렌더러에 전송된다.

기본값은 없다.

예:

```
import scaleform.gfx.*;
```

```
// m is a MovieClip on stage.  
  
DisplayObjectEx.setRendererFloat(m, 17.1717);  
trace(DisplayObjectEx.getRendererFloat(m));
```

disableBatching 정적 메서드

```
static public function disableBatching(o:DisplayObject, b:Boolean)
```

Scaleform version: 4.0.17

이 속성은 사용자 그리기를 위한 메쉬 생성 배치를 비활성화합니다.

예:

```
import scaleform.gfx.*;  
  
// m is a MovieClip on stage.  
  
DisplayObjectEx.disableBatching(batchDisable, true);  
trace(DisplayObjectEx.isBatchingDisabled(batchDisable));
```

4 FocusEventEx 익스텐션

```
package scaleform.gfx
{
    import flash.events.FocusEvent;

    public final class FocusEventEx extends FocusEvent
    {
        public var controllerIdx : uint = 0;

        public function FocusEventEx(type:String) { super(type); }
    }
}
```

표준 flash.events.FocusEvent 의 익스텐션 이벤트입니다. 이벤트를 발생시킨 컨트롤러의 0 을 기준으로 하는 인덱스를 나타내는 'controllerIdx' 멤버를 추가합니다. Extensions.enabled 속성이 true 라면 Scaleform 이 항상 표준 FocusEvent 대신 FocusEventEx 를 생성합니다. 수신된 이벤트가 FocusEventEx 의 인스턴스인지 확인할 수 있으며, 그렇다면 이벤트 객체를 익스텐션 유형으로 캐스트할 수 있습니다. 예:

```
import scaleform.gfx.*;
import flash.events.FocusEvent;

Extensions.enabled = true;

function ev(e: FocusEvent)
{
    if (e is FocusEventEx)
    {
        var ee: FocusEventEx = e as FocusEventEx;
        trace("    controllerIdx = "+ee.controllerIdx);
    }
}

stage.addEventListener(FocusEvent.MOUSE_FOCUS_CHANGE, ev);
```

controllerIdx 속성

controllerIdx : uint [read]

Scaleform 버전: 4.0.12

해당 이벤트(0 을 기준으로 하는 인덱스)에서 사용된 키보드/컨트롤러를 나타냅니다.

5 FocusManager 익스텐션

alwaysEnableArrowKeys 정적 속성

```
alwaysEnableArrowKeys:Boolean [read-write]
```

Scaleform 버전: 4.0.12

이 정적 속성을 사용하면 `_focusrect` 속성이 `false` 로 설정되더라도(포커스가 캡처되었을 때 적용됨) 화살표 키로 포커스를 바꿀 수 있습니다. 노란색 포커스 사각형이 `_focusrect = false` 를 통해 비활성화되었다면 Flash 에서 화살표 키로 포커스를 바꿀 수 없도록 기본으로 설정됩니다. 이러한 작동 방식을 변경하려면 `alwaysEnableArrowKeys` 속성을 `true` 로 설정하십시오.

disableFocusKeys 정적 속성

```
disableFocusKeys:Boolean [read-write]
```

Scaleform 버전: 4.0.12

이 정적 속성은 모든 포커스 키(TAB, Shift-TAB 과 화살표 키)의 처리를 비활성화하므로 포커스 키 관리 기능을 사용자가 직접 구현할 수 도 있습니다.

moveFocus() 정적 메서드

```
static public function moveFocus(keyToSimulate : String,  
                                startFromMovie:InteractiveObject = null,  
                                includeFocusEnabledChars : Boolean = false,  
                                controllerIdx : uint = 0) : InteractiveObject
```

Scaleform 버전: 4.0.12

TAB, Shift-TAB 또는 화살표 키 등의 포커스 키가 눌린 것처럼 가장하여 포커스 사각형을 움직이는 정적 메서드입니다. 이 메서드와 `disableFocusKeys` 속성, `modalClip` 속성으로 사용자 정의 포커스 관리를 구현할 수 있습니다.

매개변수

`keyToSimulate` : `String` - 시뮬레이트할 키 이름: "up", "down", "left", "right", "tab", "shifftab".

`startFromMovie:InteractiveObject` - `moveFocus` 가 현재 포커스된 무비클립 대신 시작 지점으로 사용할 무비클립을 지정하는 선택적 매개변수입니다. 현재 포커스된 무비클립을 시작 지점으로 사용하면 이 속성 값이 `null` 또는 `undefined` 입니다. 이렇게 하면 세 번째 선택적 매개변수를 지정하는데 유용할 수 있습니다.

`includeFocusEnabledChars` : `Boolean` - `moveFocus` 를 `focusEnabled` 속성이 지정된 무비클립 위로만 옮기거나 `tabEnabled` / `tabIndex` 속성이 지정된 무비클립 위로만 옮기도록 하는 선택적 플래그입니다. 플래그를 지정하지 않거나 `false` 로 지정하면 `tabEnabled` / `tabIndex` 속성이 지정된 무비클립만 포커스 이동에 참여합니다.

`controllerIdx` : `uint` - 작업에 사용된 컨트롤러의 인덱스입니다. 지정하지 않으면 기본 컨트롤러인 컨트롤러 0 을 사용합니다.

반환 값

다음에 포커스할 무비클립을 반환합니다. 포커스할 무비클립이 없으면 `null` 을 반환합니다.

참조:

[findFocus](#)
[disableFocusKeys](#)
[setModalClip](#)
[getModalClip](#)

findFocus() 정적 메서드

```
static public function findFocus(keyToSimulate : String,  
                                parentMovie:DisplayObjectContainer = null,  
                                loop : Boolean = false,  
                                startFromMovie:InteractiveObject = null,
```



```
includeFocusEnabledChars : Boolean = false,  
controllerIdx : uint = 0) : InteractiveObject
```

Scaleform 버전: 4.0.12

TAB, Shift-TAB 또는 화살표 키 등의 포커스 키가 눌린 것처럼 가장하여 다음으로 포커스할 항목을 찾는 데 사용하는 정적 메서드입니다. 이 메서드와 `disableFocusKeys` 익스텐션, `setModalClip/getModalClip` 익스텐션으로 사용자 정의 포커스 관리를 구현할 수 있습니다.

매개변수

`keyToSimulate : String` - 시뮬레이트할 키 이름: "up", "down", "left", "right", "tab", "shifftab".

`parentMovie:DisplayObjectContainer` - 모달 클립으로 사용하는 동영상입니다. 포커스 항목 검색은 이 클립의 자식 내에서만 수행할 수 있습니다. null 일 수 있습니다.

`loop : Boolean` - 포커스를 반복적으로 수행하기 위한 부울 플래그입니다. 예를 들어, 현재 포커스된 항목이 가장 아래에 있고 키가 "down"이면, `findFocus` 는 "null"(이 플래그가 "false"일 경우) 또는 포커스할 수 있는 항목 중 제일 적당한 항목(플래그가 "true"일 경우)을 반환합니다.

`startFromMovie:InteractiveObject` - `findFocus` 가 현재 포커스된 무비클립 대신 시작 지점으로 사용할 무비클립을 지정하는 선택적 매개변수입니다. 현재 포커스된 무비클립을 시작 지점으로 사용한다면 속성 값이 null 또는 undefined 입니다.

`includeFocusEnabledChars : Boolean` - `moveFocus` 를 `focusEnabled` 속성이 지정된 무비클립 위로만 옮기거나 `tabEnabled / tabIndex` 속성이 지정된 무비클립 위로만 옮기도록 하는 선택적 플래그입니다. 플래그를 지정하지 않거나 false 로 지정하면 `tabEnabled / tabIndex` 속성이 지정된 무비클립만 포커스 이동에 참여합니다.

`controllerIdx : uint` - 포커스를 조작하는 컨트롤러의 0 을 기준으로 하는 인덱스입니다. 이 인덱스와 포커스 그룹으로 다중 컨트롤러 포커스를 지원할 수 있습니다.

반환 값

다음에 포커스할 무비클립을 반환합니다. 포커스할 무비클립이 없으면 null 을 반환합니다.

참조:

[moveFocus](#)
[disableFocusKeys](#)
[setModalClip](#)

[getModalClip](#)

setFocus() 정적 메서드

```
static public function setFocus(obj:InteractiveObject, controllerIdx:uint = 0) : void
```

Scaleform 버전: 4.0.12

stage.focus 속성을 할당하는 것과 동일한 기능을 하는 정적 메서드입니다. 차이점이라면 해당 작업과 연결해야 할 컨트롤러의 인덱스를 지정할 수 있다는 것입니다.

매개변수

obj:InteractiveObject - 새로 포커스된 상호작용 객체입니다.

controllerIdx : uint - 해당 이벤트(0 을 기준으로 하는 인덱스)에서 사용된 키보드/컨트롤러를 나타냅니다.

getFocus() 정적 메서드

```
static public function getFocus(controllerIdx:uint = 0) : InteractiveObject
```

Scaleform 버전: 4.0.12

stage.focus 속성 값과 같은 값을 반환하는 정적 메서드입니다. 차이점이라면 해당 작업과 연결해야 할 컨트롤러의 인덱스를 지정할 수 있다는 것입니다.

매개변수

controllerIdx : uint - 해당 이벤트(0 을 기준으로 하는 인덱스)에서 사용된 키보드/컨트롤러를 나타냅니다.

반환 값

현재 포커스된 상호작용 객체를 반환합니다.

numFocusGroups 정적 속성

```
numFocusGroups() : uint [read]
```

Scaleform 버전: 4.0.12

setControllerFocusGroup 함수를 호출하도록 설정된 포커스 그룹의 개수를 반환합니다. 예를 들어 포커스 그룹 0 과 3 이 활성화되었다면 numFocusGroups 는 2 를 반환합니다.

setFocusGroupMask() 정적 메서드

```
public function setFocusGroupMask(obj:InteractiveObject, mask:uint) : void
```

Scaleform 버전: 4.0.12

무비클립과 그 무비클립의 **모든** 자식에 비트마스크를 설정하는 메서드입니다. 이 비트마스크는 무비클립에 포커스 그룹 소유권을 할당하는데, 이는 비트마스크에 표시된 컨트롤러만이 포커스를 무비클립 내부로 이동시킬 수 있음을 의미합니다. setController 와 FocusGroup 익스텐션 메서드로 포커스 그룹과 컨트롤러를 연결할 수 있습니다.

예를 들어 "button1"은 컨트롤러 0 을 통해서만 포커스를 맞출 수 있고 "movieclip2"는 컨트롤러 0 과 1 로만 맞출 수 있다면, 이 동작을 수행하려면 포커스 그룹과 컨트롤러를 연결해야 합니다.

```
FocusManager.setControllerFocusGroup(0, 0);
FocusManager.setControllerFocusGroup(1, 1);

FocusManager.setFocusGroupMask(button1, 0x1); // bit 0 - 포커스 그룹 0
FocusManager.setFocusGroupMask(movieclip2, 0x1 | 0x2); // bits 0 and 1 - 포커스
// 그룹 0 과 1
```

"focusGroupMask" 비트마스크는 부모 무비클립으로 설정할 수 있습니다. 이렇게 하면 모든 자식에게 마스크 값이 전달됩니다.

매개변수

obj:InteractiveObject - 상호작용 객체입니다.
mask:uint - 포커스 그룹 비트마스크입니다.

참조:

[setControllerFocusGroup](#)
[getFocusGroupMask](#)

getFocusGroupMask() 정적 메서드

```
static public function getFocusGroupMask(obj:InteractiveObject) : uint
```

Scaleform 버전: 4.0.12

현재 포커스 그룹 비트마스크 값(setFocusGroupMask 참조)을 반환합니다.

매개변수

obj:InteractiveObject - 상호작용 객체입니다.

반환 값

지정한 상호작용 객체의 포커스 그룹 비트마스크를 반환합니다.

참조:

[setControllerFocusGroup](#)

[setFocusGroupMask](#)

setControllerFocusGroup() 정적 메서드

```
static public function setControllerFocusGroup(controllerIdx:uint,  
                                              focusGroupId:uint) : Boolean
```

Scaleform 버전: 4.0.12

controllerIndex 가 표시한 컨트롤러와 포커스 그룹을 연결하는 정적 메서드입니다. 모든 컨트롤러는 기본적으로 포커스 그룹 0 과 연결되므로 모두 같은 포커스를 사용합니다. 하지만 컨트롤러마다 포커스를 따로 지정할 수 있습니다. 예를 들어 분할 화면 모드 등에서는 두 컨트롤러가 다른 포커스를 가져야 하므로 setControllerFocusGroup (1,1)로 컨트롤러 1 이 사용할 포커스 그룹을 만듭니다.

setControllerFocusGroup (1,0)을 호출하면 컨트롤러 0 과 1 이 다시 같은 포커스를 공유하도록 합니다.

매개변수

controllerIdx:uint - 0 을 기준으로 하는 컨트롤러 인덱스입니다.

focusGroupId:uint - 0 을 기준으로 하는 포커스 그룹 인덱스입니다.

반환 값

성공하면 true 를 반환합니다.

getControllerFocusGroup() 정적 메서드

```
static public function getControllerFocusGroup(controllerIdx:uint) : uint
```

Scaleform 버전: 4.0.12

지정한 컨트롤러와 연결된 포커스 그룹 인덱스를 반환하는 정적 메서드입니다.

매개변수

controllerIndex - 0 을 기준으로 하는 물리 컨트롤러 인덱스입니다.

반환 값

0 을 기준으로 하는 포커스 그룹 인덱스를 반환합니다.

setModalClip() 정적 메서드

```
static public function setModalClip(mc:Sprite, controllerIdx:uint = 0) : void
```

Scaleform 버전: 4.0.12

포커스 관리를 위해 지정한 동영상을 "모달" 클립으로 설정하는 정적 메서드입니다. 이는 TAB 과 Shift-TAB, 화살표 키가 "탭할 수 있는" 모든 자식에서 지정한 동영상 안에서만 포커스를 움직일 수 있다는 뜻입니다.

매개변수

controllerIdx:uint - 0 을 기준으로 하는 컨트롤러 인덱스입니다.

mc:Sprite - 모달 클립입니다.

getModalClip() 정적 메서드

```
static public function getModalClip(controllerIdx:uint = 0) : Sprite
```

Scaleform 버전: 4.0.12

지정한 컨트롤러에 대한 모달 클립을 반환하는 정적 메서드입니다.

매개변수

`controllerIdx` - 0 을 기준으로 하는 컨트롤러 인덱스입니다.

반환 값

모달 클립을 반환합니다. 모달 클립이 없으면 undefined 를 반환합니다.

6 GamePad 익스텐션

컨트롤 상수

트리거나 아날로그 스틱, 버튼과 같은 일반 게임 패드 컨트롤을 위한 보조 상수를 제공하는 클래스입니다. 이들 상수는 SF_KeyCodes.h 에서 정의한 상수와 같으며 Scaleform 이 런타임 시 올바른 값을 삽입합니다. 따라서 이들 상수를 활용하는 SWF 는 컴파일해도 제대로 작동합니다.

Scaleform FxPlayer 프레임워크는 편의상 게임 패드 컨트롤을 해당 키보드 컨트롤로 매핑합니다. 그렇지만 사용자 정의 통합이나 응용 프로그램에서 AS3 키 이벤트에 필요할 경우 이들 상수와 Gfx::Movie::HandleEvent 를 사용하여 컨트롤러 이벤트와 키보드 이벤트를 구분할 수 있습니다.

Scaleform FxPlayer 프레임워크는 실제로 이들 상수와 GamePadAnalogEvent 를 사용하여 그러한 아날로그 값에 대한 적절한 피드백을 제공합니다. GamePad 상수 대신 키보드 코드를 사용할 수도 있습니다. 게임 패드 이벤트와 키보드 이벤트를 결합할지는 개발자에게 달려 있습니다.

supportsAnalogEvents() 정적 메서드

```
public static function supportsAnalogEvents() : Boolean
```

Scaleform 버전: 4.0.13

하드웨어 플랫폼 기반용 Scaleform 버전이 트리거나 엄지 스틱 등의 게임 패드 아날로그 이벤트를 지원하면 true 를 반환합니다. 이 값으로 GamePadAnalogEvent 의 지원 여부를 확인할 수 있습니다.

참조:

[GamePadAnalogEvent](#)

7 GamePadAnalogEvent 익스텐션

```
package scaleform.gfx
{
    import flash.events.Event;

    public final class GamePadAnalogEvent extends Event
    {
        public static const CHANGE:String = "gamePadAnalogChange";

        public var code : uint = 0; // 올바른 패드 코드를 보려면
                                    // scaleform.gfx.GamePad 를 참조

        public var controllerIdx : uint = 0;
        public var xvalue : Number = 0; // 정규화된 [-1, 1]
        public var yvalue : Number = 0; // 정규화된 [-1, 1]

        public function GamePadAnalogEvent(bubbles:Boolean, cancelable:Boolean,
                                           code:uint, controllerIdx:uint = 0,
                                           xvalue:Number = 0, yvalue:Number = 0)
        {
            super(GamePadAnalogEvent.CHANGE, bubbles, cancelable);
            this.code = code;
            this.controllerIdx = controllerIdx;
            this.xvalue = xvalue;
            this.yvalue = yvalue;
        }
    }
}
```

Scaleform 이 특정 플랫폼에서 게임 패드의 아날로그 이벤트를 지원하면 이 이벤트가 발생합니다. 이 이벤트는 해당 Stage 에서만 호출하며 모든 리스너는 Stage 객체에 dispatch 되어야 합니다. Scaleform FxPlayer 프레임워크는 이들 이벤트와 'code' 속성에 맞는 GamePad 상수를 호출하지만 Scaleform 을 용도에 맞게 구현할 때 키보드 값 등 다른 값을 사용할 수도 있습니다. 예:

```
import scaleform.gfx.*;

trace("GamePadAnalogEvents supported? " + GamePad.supportsAnalogEvents());
```



```
function ev(e: GamePadAnalogEvent)
{
    trace("code = " + ev.code);
    trace("controllerIdx = " + ev.controllerIdx);
    trace("xvalue = " + ev.xvalue);
    trace("yvalue = " + ev.yvalue);
}
stage.addEventListener(GamePadAnalogEvent.CHANGE, ev);
```

코드 속성

code : uint

Scaleform 버전: 4.0.13

이 이벤트를 생성하는 데 사용한 키/컨트롤을 나타냅니다. Scaleform FxPlayer 프레임워크는 게임패드 상수를 사용합니다. [GamePad](#)를 참조하십시오.

controllerIdx 속성

controllerIdx : uint

Scaleform 버전: 4.0.13

(0 을 기준으로 하는) 이벤트에서 사용한 키보드/컨트롤러를 나타냅니다.

xvalue 속성

xvalue : Number

Scaleform 버전: 4.0.13

x 축의 현재 값을 나타냅니다. 이 값은 -1 과 1 사이(-1 과 1 포함)에서 정규화됩니다.

yvalue 속성

yvalue : Number

Scaleform 버전: 4.0.13

y 축의 현재 값을 나타냅니다. 이 값은 -1 과 1 사이(-1 과 1 포함)에서 정규화됩니다.

8 InteractiveObjectEx 익스텐션

getHitTestDisable() 정적 메서드

```
public function getHitTestDisable(o:InteractiveObject) : Boolean
```

Scaleform 버전: 4.0.13

‘hitTestDisable’ 플래그 상태를 반환합니다. true 로 설정되면 MovieClip.hitTest 함수가 충돌 감지 테스트를 할 때 이 상호작용 객체를 무시합니다. 이외에도 다른 모든 마우스 이벤트가 해당 객체로 전달되지 않습니다.

기본값은 false 입니다.

매개변수

o - 상호작용 객체입니다.

반환 값

‘hitTestDisable’ 플래그의 상태를 나타내는 Boolean 값입니다.

참조:

[InteractiveObjectEx.setHitTestDisable](#)

setHitTestDisable() 정적 메서드

```
public function setHitTestDisable(o:InteractiveObject, f:Boolean) : void
```

Scaleform 버전: 4.0.13

‘hitTestDisable’ 플래그 상태를 설정합니다. true 로 설정되면 MovieClip.hitTest 함수가 충돌 감지 테스트를 할 때 이 상호작용 객체를 무시합니다. 이외에도 다른 모든 마우스 이벤트가 해당 객체로 전달되지 않습니다. 기본값은 false 입니다.

매개변수

- o - 상호작용 객체입니다.
- f - 'hitTestDisable' 플래그의 새로운 상태를 나타내는 Boolean 값입니다.

참조:

[InteractiveObjectEx.getHitTestDisable](#)

getTopmostLevel() 정적 메서드

```
public function getTopmostLevel (o:InteractiveObject) : Boolean
```

Scaleform 버전: 4.0.13

'topmostLevel' 플래그 상태를 반환합니다. true 로 설정되면 깊이에 상관없이 이 무비클립이 다른 모든 무비클립 위에 표시됩니다.

매개변수

- o - 상호작용 객체입니다.

반환 값

'topmostLevel' 플래그 상태를 나타내는 Boolean 값을 반환합니다.

참조:

[InteractiveObjectEx.setTopmostLevel](#)

setTopmostLevel () 정적 메서드

```
public function setTopmostLevel(o:InteractiveObject, f:Boolean) : void
```

Scaleform 버전: 4.0.13

'topmostLevel' 플래그 상태를 설정합니다. true 로 설정되면 깊이에 상관없이 이 무비클립이 다른 모든 무비클립 위에 표시됩니다. 사용자 정의 마우스 커서를 모든 레벨의 객체 위에 그릴 때 유용한 매서드입니다. 기본값은 false 입니다.

여러 무비클립을 "topmostLevel"로 지정하더라도 각 무비클립의 그리기 순서는 바뀌지 않습니다. 예를 들어 A 객체가 B 객체 아래에 그려진 상태에서 A 객체의 "topmostLevel" 속성을 true 로 설정하더라도 A 객체는 B 객체 아래에 그려집니다.

참고: "topmostLevel"로 지정한 무비클립은 swapDepth ActionScript 함수의 영향을 받지 않습니다.

기본값은 false 입니다.

매개변수

- - 상호작용 객체입니다.
- £ - 'topmostLevel' 플래그의 새로운 상태를 나타내는 Boolean 값입니다.

참조:

[InteractiveObjectEx.getTopmostLevel](#)

9 KeyboardEventEx 익스텐션

```
package scaleform.gfx
{
    import flash.events.KeyboardEvent;

    public final class KeyboardEventEx extends KeyboardEvent
    {
        public var controllerIdx : uint = 0;

        public function KeyboardEventEx(type:String) { super(type); }
    }
}
```

표준 flash.events.KeyboardEvent 의 익스텐션 이벤트입니다. 이벤트를 발생시킨 컨트롤러의 0 을 기준으로 하는 인덱스를 나타내는 'controllerIdx' 멤버를 추가합니다. Extensions.enabled 속성이 true 라면 Scaleform 이 항상 표준 KeyboardEvent 대신 KeyboardEventEx 를 생성합니다. 수신된 이벤트가 KeyboardEventEx 의 인스턴스인지 확인할 수 있으며, 그렇다면 이벤트 객체를 익스텐션 유형으로 캐스트할 수 있습니다. 예:

```
import scaleform.gfx.*;
import flash.events.KeyboardEvent;

Extensions.enabled = true;

function ev(e: KeyboardEvent)
{
    if (e is KeyboardEventEx)
    {
        var ee: KeyboardEventEx = e as KeyboardEventEx;
        trace("    controllerIdx = "+ee.controllerIdx);
    }
}

stage.addEventListener(KeyboardEvent.KEY_DOWN, ev);
stage.addEventListener(KeyboardEvent.KEY_UP, ev);
```

controllerIdx 속성

```
controllerIdx : uint    [read]
```

Scaleform 버전: 4.0.12

(0 을 기준으로 하는) 이벤트에서 사용한 키보드/컨트롤러를 나타냅니다.

10 MouseCursorEvent 익스텐션

```
package scaleform.gfx
{
    import flash.events.Event;

    public final class MouseCursorEvent extends Event
    {
        public var cursor : String = "auto";
        public var mouseIdx : uint = 0;

        static public const CURSOR_CHANGE : String = "mouseCursorChange";

        public function MouseCursorEvent()
        {
            super("MouseCursorEvent", false, true);
        }
    }
}
```

이 이벤트는 마우스 커서 변경을 추적 및/또는 방지하는 역할을 수행합니다. 이 이벤트에는 다음과 같은 Event 속성 세트가 있습니다.

bubbles – false, 버블하지 않음

cancellable – true, preventDefault() 메서드를 호출하여 기본 동작(커서 변경)을 방지할 수 있습니다.

이 이벤트는 사용자 정의 애니메이션 마우스 커서를 구현하는 데 유용합니다. Scaleform 이 마우스 커서의 모양을 변경할 때마다(커서가 텍스트 상자나 버튼 위에 올려질 때 또는 flash.ui.Mouse.cursor 속성을 설정할 때) **해당 스테이지에서만**(Extensions.enabled 가 true 로 설정되었을 경우) 이 이벤트가 발생합니다. 예:

```
Extensions.enabled = true;

function e(e:MouseCursorEvent)
{
    trace(e.type + " " + e.mouseIdx + " " + e.cursor);
    e.preventDefault();
}

stage.addEventListener(MouseCursorEvent.CURSOR_CHANGE, e);
```

참고: 이 이벤트는 스테이지에 리스너가 설정된 경우에만 발생합니다.

커서 속성

`cursor : String [read]`

Scaleform 버전: 4.0.13

바뀐 커서 유형을 나타내는 속성입니다. 이 속성은 다음과 같은 `flash.ui.MouseCursor` 클래스의 문자열 값 중 하나를 포함합니다.

- `flash.ui.MouseCursor.ARROW : String = "arrow"`
- `flash.ui.MouseCursor.BUTTON : String = "button"`
- `flash.ui.MouseCursor.HAND : String = "hand"`
- `flash.ui.MouseCursor.IBEAM : String = "ibeam"`

mouseIdx 속성

`mouseIdx : uint [read]`

Scaleform 버전: 4.0.13

이벤트를 발생시킨 마우스/컨트롤러(0 을 기준으로 하는 인덱스)를 나타냅니다.

11 MouseEventEx 익스텐션

```
package scaleform.gfx
{
    import flash.events.MouseEvent;

    public final class MouseEventEx extends MouseEvent
    {
        public var mouseIdx : uint = 0;
        public var nestingIdx : uint = 0;
        public var buttonIdx : uint = 0; // LEFT_BUTTON, RIGHT_BUTTON, ...

        public static const LEFT_BUTTON : uint = 0;
        public static const RIGHT_BUTTON : uint = 1;
        public static const MIDDLE_BUTTON : uint = 2;

        public function MouseEventEx(type:String) { super(type); }
    }
}
```

표준 flash.events.MouseEvent 의 익스텐션 이벤트입니다. 다중 컨트롤러와 마우스 오른쪽/가운데 버튼을 지원하기 위한 속성을 추가합니다. Extensions.enabled 속성이 true 라면 Scaleform 이 항상 표준 MouseEvent 대신 MouseEventEx 를 생성합니다. 수신된 이벤트가 MouseEventEx 의 인스턴스인지 확인할 수 있으며, 그렇다면 이벤트 객체를 익스텐션 유형으로 캐스트할 수 있습니다. 예:

```
import scaleform.gfx.*;
Extensions.enabled = true;

stage.doubleClickEnabled = true;

function ev(e:MouseEvent):void
{
    trace("!!!! EVENT. " + cnt++);
    trace("    eventType      = "+e.type);
    trace("    bubbles          = "+e.bubbles);
    trace("    eventPhase        = "+e.eventPhase);
    trace("    target             = "+e.target.name);
    trace("    currentTarget      = "+e.currentTarget.name);
    if (e is MouseEventEx)
    {
        var ee:MouseEventEx = e as MouseEventEx;
```

```

        trace("      mouseIdx      = "+ee.mouseIdx);
        trace("      nestingIdx   = "+ee.nestingIdx);
        trace("      buttonIdx    = "+ee.buttonIdx);
    }
    trace(e);
}
stage.addEventListener(MouseEvent.CLICK, ev);
stage.addEventListener(MouseEvent.DOUBLE_CLICK, ev);
stage.addEventListener(MouseEvent.MOUSE_DOWN, ev);
stage.addEventListener(MouseEvent.MOUSE_UP, ev);
stage.addEventListener(MouseEvent.CLICK, ev);
stage.addEventListener(MouseEvent.DOUBLE_CLICK, ev);

```

참고: 익스텐션이 활성화되면 오른쪽, 가운데, 중앙 등 마우스 버튼 이벤트를 위한 마우스 이벤트가 발생합니다. 이때 buttonIdx 속성을 확인해 이벤트를 발생시킨 버튼을 알아내야 합니다.

buttonIdx 속성

buttonIdx : uint [read]

Scaleform 버전: 4.0.13

이벤트를 발생시킨 버튼(0 을 기준으로 하는 인덱스)를 나타냅니다. 그 값은 다음 중 하나입니다.

- MouseEventEx.LEFT_BUTTON : uint = 0 - 마우스 왼쪽 버튼
- MouseEventEx.RIGHT_BUTTON : uint = 1 - 마우스 오른쪽 버튼
- MouseEventEx.MIDDLE_BUTTON : uint = 2 - 마우스 가운데 버튼
- 버튼이 세 개 이상 달린 마우스라면 2 이상의 값도 유효합니다.

mouseIdx 속성

mouseIdx : uint [read]

Scaleform 버전: 4.0.13

이벤트를 발생시킨 마우스/컨트롤러(0 을 기준으로 하는 인덱스)를 나타냅니다.

nestingIdx 속성

nestingIdx : uint [read]

Scaleform 버전: 4.0.13

이 속성은 rollOver/Out 이나 mouseOver/Out, dragOver/Out 이벤트에서 선택적입니다. 이 매개변수는 같은 무비클립에 중첩된 rollover/dragover 이벤트의 인덱스를 지정합니다.

rollOver/rollOut 나 mouseOver/Out, dragOver/dragOut 이벤트가 마우스 커서마다 따로 중첩되어 생성되면 이 매개변수가 중첩의 0 을 기준으로 하는 인덱스를 나타냅니다. 초기 이벤트의 매개변수는 0 이며 두 번째 커서가 같은 무비클립 위로 이동하면 멤버가 1 로 설정되고 두 번째 rollOver/rollOut 이벤트가 발생합니다. 커서 중 하나라도 무비클립을 벗어나면 멤버가 1 로 설정되고 rollOut/mouseOut/dragOut 이 발생하며 멤버가 0 으로 설정된 다음 마지막 rollOut/mouseOut/dragOut 이벤트가 발생합니다.

12 TextEventEx 익스텐션

```
package scaleform.gfx
{
    import flash.events.TextEvent;

    public final class TextEventEx extends TextEvent
    {
        public var controllerIdx : uint = 0;

        public function TextEventEx(type:String) { super(type); }
    }
}
```

표준 flash.events.TextEvent 의 익스텐션 이벤트입니다. 이벤트를 발생시킨 컨트롤러의 0 을 기준으로 하는 인덱스를 나타내는 'controllerIdx' 멤버를 추가합니다. Extensions.enabled 속성이 true 라면 Scaleform 이 항상 표준 TextEvent 대신 TextEventEx 를 생성합니다. 수신된 이벤트가 TextEventEx 의 인스턴스인지 확인할 수 있으며, 그렇다면 이벤트 객체를 익스텐션 유형으로 캐스트할 수 있습니다. 예:

```
import scaleform.gfx.*;
import flash.events.TextEvent;

Extensions.enabled = true;

function ev(e: TextEvent)
{
    if (e is TextEventEx)
    {
        var ee: TextEventEx = e as TextEventEx;
        trace("    controllerIdx = "+ee.controllerIdx);
    }
}

txf.addEventListener(TextEvent.TEXT_INPUT, ev);
```

controllerIdx 속성

controllerIdx : uint [read]

Scaleform 버전: 4.0.12

(0 을 기준으로 하는) 이벤트에서 사용한 키보드/컨트롤러를 나타냅니다.

LINK_MOUSE_OVER/LINK_MOUSE_OUT 이벤트

```
public static const LINK_MOUSE_OVER:String = "linkMouseOver";  
public static const LINK_MOUSE_OUT:String = "linkMouseOut";
```

Scaleform 버전: 4.0.14

TextFieldEx.LINK_MOUSE_OVER 와 TextFieldEx.LINK_MOUSE_OUT 이벤트 익스텐션을 사용하여 TextField 링크(HTML 태그로 지정)에서 mouse over 나 mouse out 이벤트를 감지할 수 있습니다. 이들 이벤트는 기타 AS3 이벤트와 동일하게 작동하며 addEventListener 를 사용하여 감지할 수 있습니다.

13 시스템 익스텐션

actionVerbose 정적 속성

`actionVerbose:Boolean` [read-write]

Scaleform 버전: 4.0.12

조작 부호 추적을 활성화하거나 비활성화합니다.

getStackTrace() 정적 메서드

`public function getStackTrace() : String`

Scaleform 버전: 4.0.12

문자열로 된 스택 Trace 를 얻습니다.

getCodeFileName() 정적 메서드

`public function getCodeFileName() : String`

Scaleform 버전: 4.0.12

실행하고 있는 코드의 파일 이름을 얻습니다.

14 TextFieldEx 익스텐션

appendHtml() 정적 메서드

```
public function appendHtml(textField:TextField, newHtml:String) : void
```

Scaleform 버전: 4.0.12

newHtml 매개변수에서 지정한 HTML 을 텍스트 필드의 텍스트 끝에 추가합니다. 이 메서드는 `txt.htmlText += moreHtml` 와 같은 `htmlText` 속성에서 증가 연산자(+=)보다 효율적입니다. `htmlText` 속성에서 일반 += 연산자를 사용하면 HTML 문자열을 생성하고 HTML 에 새로운 부분을 추가한 다음 HTML 전체 구문을 처음부터 다시 분석합니다. 이 함수는 증분 HTML 구문분석(예: newHtml 문자열 매개변수의 HTML 만을 구문분석함)을 수행합니다. 그러므로 newHtml 매개변수에 포함된 HTML 을 형식에 맞게 잘 짜야 합니다. += 연산자를 사용하면 그럴 필요가 없습니다. 이는 콘텐츠가 대량으로 포함된 텍스트 필드에서 특히 중요합니다.

참고: 텍스트 필드에 스타일 시트가 적용되었다면 이 메서드는 작동하지 않습니다.

매개변수

`textField:TextField` - HTML 의 끝에 추가할 textfield 입니다.

`newHtml:String` - 기존 텍스트에 추가할 HTML 문자열입니다.

setIMEEnabled() 정적 메서드

```
static public function setIMEEnabled(textField:TextField, isEnabled:Boolean): void
```

Scaleform 버전: 4.0.12

지정한 textfield 의 IME 를 활성화하거나 비활성화합니다. isEnabled 를 false 로 설정하면 이 텍스트 필드에서는 IME 가 활성화되지 않습니다. IME 는 기본적으로 활성화되어 있습니다.

매개변수

`textField` : `TextField` - 작업할 `textfield` 입니다.

`isEnabled` : `Boolean` - `true` 라면 IME 가 활성화되고 그렇지 않다면 비활성화됩니다.

`setVerticalAlign()` 정적 메서드

```
public function setVerticalAlign(textField:TextField, valign:String) : void
```

Scaleform 버전: 4.0.12

텍스트 상자에 포함된 텍스트의 수직 정렬을 설정합니다. 이 속성의 유효 값은 `TextFieldEx` 클래스에 정의된 다음과 같은 상수들입니다.

```
public static const VALIGN_TOP:String      = "top";  
public static const VALIGN_CENTER:String   = "center";  
public static const VALIGN_BOTTOM:String   = "bottom";
```

이 속성을 `center` 로 설정하면 텍스트 상자의 텍스트가 가운데로 정렬되고 `bottom` 으로 설정하면 텍스트가 하단으로 정렬됩니다(아래 그림 참조).



기본값은 `top` 입니다.

매개변수

`textField:TextField` - 수직 정렬을 설정할 `textfield` 입니다.

`valign:String` - 정렬 값 ("`top`", "`center`", "`bottom`").