



# **iSpring SDK Guide**

# iSpring SDK Guide

## Table of contents

|   |    |
|---|----|
| 1 iSpring SDK Guide.....                                      | 8  |
| 1.1 What's new in iSpring SDK.....                            | 8  |
| 1.2 Animation Effects Support.....                            | 10 |
| 1.3 iSpring SDK COM API.....                                  | 14 |
| 1.3.1 iSpring.AppearanceSettings Interface.....               | 14 |
| 1.3.2 iSpring.AudioBitrateEnum Enumeration.....               | 17 |
| 1.3.3 BackgroundAudioSettings Interface.....                  | 18 |
| 1.3.4 iSpring.BuiltInPluginType Enumeration.....              | 18 |
| 1.3.5 iSpring.Company Interface.....                          | 19 |
| 1.3.6 iSpring.CorePlugin Interface.....                       | 19 |
| 1.3.7 iSpring.CorePlugins Interface.....                      | 20 |
| 1.3.8 iSpring.FlashVideoFormat Enumeration.....               | 21 |
| 1.3.9 iSpring.Image Interface.....                            | 21 |
| 1.3.10 iSpring.ImageFileType Enumeration.....                 | 22 |
| 1.3.11 iSpring.MediaSettings Interface.....                   | 22 |
| 1.3.12 iSpring.MouseClickAdvanceType Enumeration.....         | 26 |
| 1.3.13 iSpring.NavigationSettings Interface.....              | 26 |
| 1.3.14 iSpring.PasswordSettings Interface.....                | 27 |
| 1.3.15 iSpring.PlaybackSettings Interface.....                | 28 |
| 1.3.16 iSpring.PlayerSettings Interface.....                  | 30 |
| 1.3.17 iSpring.Plugins Interface.....                         | 30 |
| 1.3.18 iSpring.Presentation Interface.....                    | 31 |
| 1.3.19 PresentationConverter Class.....                       | 33 |
| 1.3.19.1 PresentationConverter.CancelConversion() Method..... | 33 |

|   |    |
|---|----|
| 1.3.19.2 LastErrorDescription Property.....                 | 35 |
| 1.3.19.3 LicenseName Property.....                          | 36 |
| 1.3.19.4 LicenseNumber Property.....                        | 38 |
| 1.3.19.5 PresentationOpened Property.....                   | 39 |
| 1.3.19.6 Version Property.....                              | 40 |
| 1.3.20 iSpring.Presenter Interface.....                     | 41 |
| 1.3.21 iSpring.Presenters Interface.....                    | 42 |
| 1.3.22 iSpring.ProtectionSettings Interface.....            | 43 |
| 1.3.23 iSpring.Publish Interface.....                       | 44 |
| 1.3.24 iSpring.Settings Interface.....                      | 45 |
| 1.3.25 iSpring.Skin Interface.....                          | 49 |
| 1.3.26 iSpring.Skins Interface.....                         | 50 |
| 1.3.27 iSpring.SkinSettings Interface.....                  | 52 |
| 1.3.28 iSpring.Slide Interface.....                         | 53 |
| 1.3.29 iSpring.Slides Interface.....                        | 54 |
| 1.3.30 iSpring.VideoConverter Interface.....                | 55 |
| 1.3.31 iSpring.VideoConversionResult Enumeration.....       | 56 |
| 1.3.32 iSpring.VideoObjectInfo Interface.....               | 56 |
| 1.3.33 iSpring.VideoObjectPlaybackMode Enumeration.....     | 58 |
| 1.3.34 iSpring.VideoObjectProcessingResult Enumeration..... | 59 |
| 1.3.35 iSpring.VideoObjectType Enumeration.....             | 59 |
| 1.3.36 iSpring.VideoParameters Interface.....               | 59 |
| 1.3.37 iSpring.VideoPlugins Interface.....                  | 60 |
| 1.3.38 iSpring.Watermark Interface.....                     | 61 |
| 1.3.39 iSpring.WatermarkAlignment Enumeration.....          | 63 |
| 1.4 iSpring SDK command line interface.....                 | 63 |
| 1.5 ActionScript API References.....                        | 78 |
| 1.5.1 iSpring Presentation structure.....                   | 78 |
| 1.5.1.1 Presentation assets.....                            | 79 |

|  |     |
|--|-----|
| 1.5.1.2 Player core module.....  | 79  |
| 1.5.1.3 Skin module.....   | 80  |
| 1.5.1.4 References.....  | 81  |
| 1.5.2 Flash Presentation ActionScript API Reference.....               | 81  |
| 1.5.2.1 fsplayer.api.CPresentationLoader Class.....                    | 81  |
| 1.5.2.2 fsplayer.api.IAnimationStep Interface.....                     | 82  |
| 1.5.2.2.1 Remarks.....   | 83  |
| 1.5.2.3 fsplayer.api.IAnimationSteps Interface.....                    | 84  |
| 1.5.2.4 fsplayer.api.ICompanyInfo Interface.....                       | 86  |
| 1.5.2.5 fsplayer.api.ICompanyLogo Interface.....                       | 86  |
| 1.5.2.6 fsplayer.api.ICompanyLogoLoadingListener Interface.....        | 87  |
| 1.5.2.7 fsplayer.api.IPlaybackListener Interface.....                  | 88  |
| 1.5.2.8 fsplayer.api.IPlayer Interface.....                            | 91  |
| 1.5.2.8.1 Remarks.....   | 92  |
| 1.5.2.9 fsplayer.api.IPlayerCore Interface.....                        | 93  |
| 1.5.2.10 fsplayer.api.IPlayerEx Interface.....                         | 93  |
| 1.5.2.11 fsplayer.api.IPlayerListener Interface.....                   | 94  |
| 1.5.2.12 fsplayer.api.IPlayerListenerEx Interface.....                 | 95  |
| 1.5.2.13 fsplayer.api.IPresentationInfo Interface.....                 | 96  |
| 1.5.2.14 fsplayer.api.IPresentationPlaybackController Interface.....   | 97  |
| 1.5.2.15 fsplayer.api.IPresentationPlaybackControllerEx Interface..... | 100 |
| 1.5.2.16 fsplayer.api.IPresenterInfo Interface.....                    | 101 |
| 1.5.2.17 fsplayer.api.IPresenterPhoto Interface.....                   | 102 |
| 1.5.2.17.1 Remarks.....  | 102 |
| 1.5.2.18 fsplayer.api.IPresenterPhotoLoadingListener Interface.....    | 102 |
| 1.5.2.19 fsplayer.api.IPresentersCollection Interface.....             | 103 |
| 1.5.2.20 fsplayer.api.IPresenterVideo Interface.....                   | 103 |
| 1.5.2.21 fsplayer.api.IPresenterVideoController Interface.....         | 104 |
| 1.5.2.22 fsplayer.api.IPresenterVideoListener Interface.....           | 104 |

|   |     |
|---|-----|
| 1.5.2.23 fsplayer.api.IReferenceInfo Interface.....                         | 105 |
| 1.5.2.24 fsplayer.api.IReferencesCollection Interface.....                  | 105 |
| 1.5.2.25 fsplayer.api.ISlideInfo Interface.....                             | 106 |
| 1.5.2.25.1 Remarks.....   | 108 |
| 1.5.2.26 fsplayer.api.ISlidesCollection Interface.....                      | 108 |
| 1.5.2.27 fsplayer.api.ISlideThumbnail Interface.....                        | 109 |
| 1.5.2.27.1 Remarks.....   | 109 |
| 1.5.2.28 fsplayer.api.ISlideThumbnailLoadingListener Interface.....         | 110 |
| 1.5.2.29 fsplayer.api.ISlideResources Interface.....                        | 110 |
| 1.5.2.30 fsplayer.api.ISoundController Interface.....                       | 111 |
| 1.5.2.30.1 Remarks.....   | 112 |
| 1.5.2.31 fsplayer.api.ISoundListener Interface.....                         | 112 |
| 1.5.2.32 fsplayer.api.IThumbnailsCollection Interface.....                  | 113 |
| 1.5.2.33 ActionScript 3 and Flex 2 support.....                             | 114 |
| 1.5.2.33.1 iSpring.as2player.AcquireFocusEvent Class.....                   | 114 |
| 1.5.2.33.2 iSpring.as2player.IAnimationStep Interface.....                  | 115 |
| 1.5.2.33.3 iSpring.as2player.IAnimationSteps Interface.....                 | 117 |
| 1.5.2.33.4 iSpring.as2player.ICompanyInfo Interface.....                    | 118 |
| 1.5.2.33.5 iSpring.as2player.ICompanyLogo Interface.....                    | 119 |
| 1.5.2.33.6 iSpring.as2player.IPlayer Interface.....                         | 120 |
| 1.5.2.33.7 iSpring.as2player.IPresentationInfo Interface.....               | 122 |
| 1.5.2.33.8 iSpring.as2player.IPresentationPlaybackController Interface..... | 124 |
| 1.5.2.33.9 iSpring.as2player.IPresenterInfo Interface.....                  | 128 |
| 1.5.2.33.10 iSpring.as2player.IPresentersCollection Interface.....          | 128 |
| 1.5.2.33.11 iSpring.as2player.IPresenterVideo Interface.....                | 129 |
| 1.5.2.33.12 iSpring.as2player.IReferenceInfo Interface.....                 | 129 |
| 1.5.2.33.13 iSpring.as2player.IReferencesCollection Interface.....          | 130 |
| 1.5.2.33.14 iSpring.as2bridge.ISlideInfo Interface.....                     | 131 |
| 1.5.2.33.15 iSpring.as2bridge.ISlideResources Interface.....                | 133 |

|             |  |     |
|-------------|--|-----|
| 1.5.2.33.16 | iSpring.as2player.ISlidesCollection Interface.....   | 133 |
| 1.5.2.33.17 | iSpring.as2player.ISoundController Interface.....    | 134 |
| 1.5.2.33.18 | iSpring.as2player.PlaybackEvent Class.....           | 135 |
| 1.5.2.33.19 | iSpring.as2player.PlaybackPositionEvent Class.....   | 136 |
| 1.5.2.33.20 | iSpring.as2player.PlayerEvent Class.....             | 136 |
| 1.5.2.33.21 | iSpring.as2player.PresentationLoader Class.....      | 137 |
| 1.5.2.33.22 | iSpring.as2player.SlidePlaybackEvent Class.....      | 138 |
| 1.5.2.33.23 | iSpring.as2player.SoundEvent Class.....              | 139 |
| 1.5.2.33.24 | iSpring.as2player.StepPlaybackEvent Class.....       | 140 |
| 1.5.2.33.25 | Obsolete ActionScript 3 API.....                     | 141 |
| 1.5.3       | Skin Development Reference.....                      | 162 |
| 1.5.3.1     | fsplayer.ui.skins.ISkin Interface.....               | 162 |
| 1.5.3.1.1   | Remarks.....   | 162 |
| 1.5.3.1.2   | Samples.....   | 162 |
| 1.5.3.2     | fsplayer.ui.skins.ISkinListener Interface.....       | 165 |
| 1.5.3.3     | fsplayer.ui.skins.ISkinSizeController Interface..... | 165 |
| 1.5.3.4     | fsplayer.ui.skins.ISlideShowWindow Interface.....    | 166 |
| 1.5.4       | Flash Slide ActionScript API Reference.....          | 167 |
| 1.5.4.1     | slideplayer.api.IHyperlinksManager Interface.....    | 167 |
| 1.5.4.2     | slideplayer.api.IPlaybackController Interface.....   | 170 |
| 1.5.4.3     | slideplayer.api.IPlaybackListener Interface.....     | 173 |
| 1.5.4.4     | slideplayer.api.ISlideController Interface.....      | 174 |
| 1.5.4.5     | slideplayer.api.ISoundController Interface.....      | 176 |
| 1.6         | JavaScript API Reference.....                        | 178 |
| 1.6.1       | fsplayer.api.Player Class.....                       | 178 |
| 1.6.2       | iSpring.as2player.PresentationLoader Class.....      | 179 |
| 1.6.3       | fsplayer.api.PlaybackController Class.....           | 180 |
| 1.6.4       | fsplayer.api.PlaybackControllerEvents Class.....     | 183 |
| 1.6.5       | fsplayer.api.PresentationInfo Class.....             | 184 |

|  |     |
|--|-----|
| 1.6.6 fsplayer.api.SlidesCollection Class.....     | 185 |
| 1.6.7 fsplayer.api.PresentersCollection Class..... | 185 |
| 1.6.8 fsplayer.api.SlideInfo Class.....            | 185 |
| 1.7 Support.....                                   | 186 |

## 1. iSpring SDK Guide

### 1.1. What's new in iSpring SDK

New features in iSpring SDK 4.2

- Youtube videos support
- Preserving slide notes formatting
- Advanced Glossy player template
- Ability to make slide background transparent
- Ability to add a custom watermark to the generated presentations
- PowerPoint to Flash conversion core improvements
- Ability to make slide background transparent
- Ability to add a custom watermark to the generated presentations
- PowerPoint to Flash conversion core improvements

New features in iSpring SDK 3.5

- Animations by words and letters
- Repeated Animations
- Improved 3D objects positioning
- Enhanced Advanced Player
- Improved parallel conversion
- New [ISkinSizeController Interface](#) in ActionScript API

New features in iSpring SDK 3.2

- PowerPoint 2007 presentations support
- [ActionScript 3 and Flex support](#)
- Flash presentation [preloader](#)
- [EXE file creation](#)
- Ability to detach video, Flash and audio resources from the generated Flash presentation
- A set of new skins
- iSpring SDK Code Builder
- Presentation autostart control
- New enhanced [COM API](#)
- iSpring SDK Java wrapper
- New C++ samples
- New C# samples
- New VB.NET samples

New features in FlashSpring Server 2.4

**New features in Server COM API**

- Control over the smoothing of raster images.
- Settings for flexible control over the compression of GIF and JPEG images.

**New PowerPoint to Flash conversion facilities**

- Improved conversion quality of grouped shapes
- Improved WMA and MP3 audio files processing
- Improved processing of charts and diagrams
- Improved processing of embedded Flash objects

New in FlashSpring Server 2.3

**New features in Server COM API**

- New methods and interfaces in COM API for conversion presentations in different output modes
- Optimization of Flash movie to the screen resolution
- Opening hyperlinks in specified browser window
- New samples in **Java** language - you can find them in **samples\server-side-samples\java** subfolder of iSpring SDK installation directory.

**New features in Client side API**

- [ActionScript API](#) enhancements for handy playback control by animation steps.

**New PowerPoint to Flash conversion facilities**

- Right-to-Left languages support
- Synchronization of animations and sounds
- Special skin for banners creation
- Disable "Print..." option
- Looped video support
- Conversion of large (300+ Mb) presentations
- High quality processing of grouped shapes
- Vector conversion of tables
- Support of animated GIF images
- Improved processing of hyperlinks on Master Slides

New in FlashSpring Server 2.1

- Slide navigation by **animation steps and positions**
- Player color scheme customization
- Presentation viewing restriction by domain name
- Improved support of **Master Slides'** objects animations and interactivity
- Improved **grouped shapes** positioning
- **.WMA audio** files support
- **Looped sounds** for animation **effects** and slide **transitions**
- **--allowed-domain** option is available through [command line](#)

#### New in FlashSpring Server 2.0

- 180+ PowerPoint 2003 animation effects support
- Full support of slide transition effects
- Vector representation of basic PowerPoint shapes
- Embedded video clips support
- [COM API](#) for use in Visual Basic, C#, C++, PHP and other development environments
- Extended [ActionScript API](#) for presentation playback control and custom skin development
- Slide thumbnails generation
- Zipped output
- Background audio for whole presentation
- Background audio for separate slides
- Loading and saving conversion settings in XML files
- Picture-style bullets support
- Samples of server-side applications in ASP.NET, C#, VBScript, and PHP
- Samples for command line
- Samples of client-side applications with the use of ActionScript API

Find out more about the history of the product development on [iSpring web site](#).

## 1.2. Animation Effects Support

iSpring PRESENTER supports 180+ animation effects of PowerPoint 2003 and PowerPoint 2007 as well as combinations of effects, effect enhancements & timing and all possible directions for supported effects.

iSpring PRESENTER smoothly converts repeated animations with set number of strokes, animations with "Auto-reverse" and "Rewind at end" settings.

Animation effects are listed below in the same way as they appear in PowerPoint. Effects that are not supported in current iSpring version are highlighted with light grey color.

















| Entrance Effects | Exit Effects |
|------------------|--------------|
|------------------|--------------|

|   |  |
|---|--|
| <p><b>Basic</b></p> <ul style="list-style-type: none"> <li> Appear</li> <li> Box</li> <li> Circle</li> <li> Diamond</li> <li> Flash Once</li> <li> Peek In</li> <li> Random Bars</li> <li> Split</li> <li> Wedge</li> <li> Wipe</li> <li> Blinds</li> <li> Checkerboard</li> <li> Crawl In</li> <li> Dissolve In</li> <li> Fly In</li> <li> Plus</li> <li> Random Effects</li> <li> Strips</li> <li> Wheel</li> </ul> | <p><b>Basic</b></p> <ul style="list-style-type: none"> <li> Blinds</li> <li> Checkerboard</li> <li> Crawl Out</li> <li> Disappear</li> <li> Flash Once</li> <li> Peek Out</li> <li> Random Bars</li> <li> Split</li> <li> Wedge</li> <li> Wipe</li> <li> Box</li> <li> Circle</li> <li> Diamond</li> <li> Dissolve Out</li> <li> Fly Out</li> <li> Plus</li> <li> Random Effects</li> <li> Strips</li> <li> Wheel</li> </ul> |
| <p><b>Subtle</b></p> <ul style="list-style-type: none"> <li> Expand</li> <li> Faded Swivel</li> <li> Fade</li> <li> Faded Zoom</li> </ul> <p><b>Moderate</b></p> <ul style="list-style-type: none"> <li> Ascend</li> <li> Color Typewriter</li> <li> Descend</li> <li> Grow &amp; Turn</li> <li> Spinner</li> <li> Unfold</li> <li> Center Revolve</li> <li> Compress</li> <li> Ease In</li> <li> Rise Up</li> <li> Stretch</li> <li> Zoom</li> </ul>   | <p><b>Subtle</b></p> <ul style="list-style-type: none"> <li> Contract</li> <li> Faded Swivel</li> <li> Fade</li> <li> Faded Zoom</li> </ul> <p><b>Moderate</b></p> <ul style="list-style-type: none"> <li> Ascend</li> <li> Collapse</li> <li> Descend</li> <li> Grow &amp; Turn</li> <li> Spinner</li> <li> Unfold</li> <li> Center Revolve</li> <li> Color Typewriter</li> <li> Ease Out</li> <li> Sink Down</li> <li> Stretchy</li> <li> Zoom</li> </ul>  |

| Exciting    |          | Exciting    |            |
|-------------|----------|-------------|------------|
| Boomerang   | Bounce   | Boomerang   | Bounce     |
| Credits     | Curve Up | Credits     | Curve Down |
| Flip        | Float    | Flip        | Float      |
| Fold        | Glide    | Fold        | Glide      |
| Light Speed | Magnify  | Light Speed | Magnify    |
| Pinwheel    | Sling    | Pinwheel    | Sling      |
| Spiral In   | Swish    | Spiral Out  | Swish      |
| Swivel      | Thread   | Swivel      | Thread     |
| Whip        |          | Whip        |            |

| Emphasis Effects  | Motion Paths |
|---|--------------|
| <p><b>Basic</b></p> <ul style="list-style-type: none"> <li> Change Fill Color</li> <li> Change Font Color</li> <li> Change Font Style</li> <li> Grow/Shrink</li> <li> Transparency</li> <li> Change Font</li> <li> Change Font Size</li> <li> Change Line Color</li> <li> Spin</li> </ul> <p><b>Subtle</b></p> <ul style="list-style-type: none"> <li> Bold Flash</li> <li> Brush On Underline</li> <li> Color Wave</li> <li> Complementary Color 2</li> <li> Darken</li> <li> Flash Bulb</li> <li> Vertical Highlight</li> <li> Brush On Color</li> <li> Color Blend</li> <li> Complementary Color</li> <li> Contrasting Color</li> <li> Desaturate</li> <li> Lighten</li> </ul> |              |

|  |   |
|--|---|
|  | <p><b>Basic</b></p> <ul style="list-style-type: none"> <li> 4 Point Star</li> <li> 5 Point Star</li> <li> 6 Point Star</li> <li> 8 Point Star</li> <li> Circle</li> <li> Crescent Moon</li> <li> Diamond</li> <li> Equal Triangle</li> <li> Football</li> <li> Heart</li> <li> Hexagon</li> <li> Octagon</li> <li> Parallelogram</li> <li> Pentagon</li> <li> Right Triangle</li> <li> Square</li> <li> Teardrop</li> <li> Trapezoid</li> </ul> <p><b>Lines &amp; Curves</b></p> <ul style="list-style-type: none"> <li> Arc Down</li> <li> Arc Left</li> <li> Arc Right</li> <li> Arc Up</li> <li> Bounce Left</li> <li> Bounce Right</li> </ul> |
| <p><b>Moderate</b></p> <ul style="list-style-type: none"> <li> Flicker</li> <li> Grow With Color</li> <li> Shimmer</li> <li> Teeter</li> </ul> <p><b>Exciting</b></p> <ul style="list-style-type: none"> <li> Blast</li> <li> Blink</li> <li> Bold Reveal</li> <li> Style Emphasis</li> <li> Wave</li> </ul> | <ul style="list-style-type: none"> <li> Curvy Left</li> <li> Curvy Right</li> <li> Decaying Wave</li> <li> Diagonal Down Right</li> <li> Diagonal Up Right</li> <li> Down</li> <li> Funnel</li> <li> Heartbeat</li> <li> Left</li> <li> Right</li> <li> S Curve 1</li> <li> S Curve 2</li> <li> Sine Wave</li> <li> Spiral Left</li> <li> Spiral Right</li> <li> Spring</li> <li> Stairs Down</li> <li> Turn Down</li> <li> Turn Down Right</li> <li> Turn Up</li> <li> Turn Up Right</li> <li> Up</li> <li> Wave</li> <li> Zigzag</li> </ul>   |

| Special   |   |
|---|---|
|  Bean                |  Buzz Saw          |
|  Curved Square       |  Curved X          |
|  Curvy Star          |  Figure 8 Four     |
|  Horizontal Figure 8 |  Inverted Square   |
|  Inverted Triangle   |  Loop de Loop      |
|  Neutron             |  Peanut            |
|  Plus                |  Pointy Star       |
|  Swoosh              |  Vertical Figure 8 |

## 1.3. iSpring SDK COM API

### 1.3.1. iSpring.AppearanceSettings Interface

The **AppearanceSettings** Interface represents appearance settings of the generated Flash content.

| Properties      |            |            |               |  |
|-----------------|------------|------------|---------------|--|
| Property        | Value Type | Access     | Default value | Property description   |
| AddPreloader    | bool       | Read/write | false         | A Boolean value indicating whether a preloading module is added to the generated Flash presentation.<br><b>Note:</b> the preloading module is not added when the presentation is exported into a set of standalone Flash slides. |
| BackgroundColor | uint       | Read/write | 0xFFFFFFFF    | Background color   |

|                 |       |            |       |  |
|-----------------|-------|------------|-------|--|
|                 |       |            |       | of the generated Flash presentation in 0xBBGGRR format:<br>color = Red + (Green * 255) + (Blue * 65536)  |
| DisablePrinting | bool  | Read/write | false | A Boolean value indicating whether the "Print..." option of the Adobe Flash Player context menu is disabled.<br><b>Note:</b> if the <b>DisableMenu</b> property is set to true, the <b>DisablePrinting</b> property value doesn't matter |
| DisableMenu     | bool  | Read/write | false | A Boolean value indicating whether the Adobe Flash Player context menu is disabled.  |
| FrameRate       | float | Read/write | 24    | A frame rate of the generated Flash presentation.<br><b>Note:</b> a frame rate value must be in range: [0.1; 100]  |
| FullScreen      | bool  | Read/write | false | A Boolean value indicating whether the generated Flash presentation occupies a whole screen area.<br><b>Note:</b> there are some <a href="#">full-screen mode restrictions</a> for Flash files running in browser                        |

|                    |      |            |       |   |
|--------------------|------|------------|-------|---|
|                    |      |            |       | <p>window:</p> <ul style="list-style-type: none"> <li>• Adobe Flash Player 9.0.28.0 or later is required</li> <li>• allowFullScreen tag parameter must be included to the &lt;object&gt; and &lt;embed&gt; tags of the HTML code.</li> <li>• Users cannot enter text in text input fields while in full-screen mode.</li> </ul> |
| OptimizeImagesForS | bool | Read/write | false | <p>A Boolean value indicating whether the generated Flash content will be optimized for the particular screen resolution.<br/> <b>Note:</b> the ScreenHeight and the ScreenWidth properties specify height and width of the screen the generated presentation will be optimized for</p>   |
| ScreenHeight       | int  | Read/write | 768   | <p>The screen height images will be optimized for.<br/> <b>Note:</b> this property is used if the <b>OptimizeImagesForScreenResolution</b> property is set to true</p>  |

|                       |      |            |      |  |
|-----------------------|------|------------|------|--|
| ScreenWidth           | int  | Read/write | 1024 | The screen width images will be optimized for.<br><b>Note:</b> this property is used if the <b>OptimizeImagesForScreenResolution</b> property is set to true |
| SmoothImages          | bool | Read/write | true | A Boolean value indicating whether raster images smoothing is turned on  |
| ExportSlideBackground | bool | Read/write | true | Indicates whether the slide background should be exported  |

**See also**

[PresentationConverter Class](#)

[Settings Interface](#)

[iSpring SDK COM API](#)

### 1.3.2. iSpring.AudioBitrateEnum Enumeration

The **AudioBitrateEnum** Enumeration defines the possible values of the [AudioBitrate](#) property of the [MediaSettings](#) Interface. Possible values are:

- AB\_ORIGINAL** = 0,
- AB\_8** = 8,
- AB\_16** = 16,
- AB\_24** = 24,
- AB\_32** = 32,
- AB\_40** = 40,
- AB\_48** = 48,
- AB\_56** = 56,
- AB\_64** = 64,
- AB\_80** = 80,
- AB\_96** = 96,
- AB\_112** = 112,

**AB\_128** = 128,  
**AB\_160** = 160,  
**AB\_192** = 192,  
**AB\_224** = 224,  
**AB\_256** = 256,  
**AB\_320** = 320

**See also**

[MediaSettings Interface](#)  
[iSpring SDK COM API](#)

### 1.3.3. BackgroundAudioSettings Interface

The **BackgroundAudioSettings** Interface provides Flash presentation background audio settings.

| Properties |            |            |               |   |
|------------|------------|------------|---------------|---|
| Property   | Value Type | Access     | Default value | Property description  |
| LoopAudio  | bool       | Read/write | false         | Indicates whether the presentation background audio is looped |
| SourceFile | string     | Read/write | ""            | Presentation background audio file path                       |

**See also**

[Presentation Interface](#)  
[Slide.BackgroundAudioFile Property](#)

### 1.3.4. iSpring.BuiltInPluginType Enumeration

**BuiltInPluginType** Enumeration defines the possible values of the [AddBuiltInPlugin](#) method parameter of the [CorePlugins](#) interface. Possible values are:

**JAVASCRIPT\_CONNECTOR** = 0,  
**ACTIONSCRIPT3\_CONNECTOR** = 1

**See also**

[CorePlugins Interface](#)  
[iSpring SDK COM API](#)

### 1.3.5. iSpring.Company Interface

The **Company** interface provides information about presenter company.

| Properties |                       |            |               |  |
|------------|-----------------------|------------|---------------|--|
| Property   | Value Type            | Access     | Default value | Property description   |
| HasLogo    | bool                  | Read only  |               | Indicates whether the company has logo   |
| Logo       | <a href="#">Image</a> | Read only  |               | Returns the <a href="#">Image</a> interface providing information about company logotype |
| WebSite    | string                | Read/Write |               | Company website address  |

| Methods                                     |   |
|---|---|
| Method                                      | Method description  |
| <a href="#">Image</a> LoadLogo(string path) | Loads company logo from the specifies image file and returns the <a href="#">Image</a> interface of the newly loaded logo |
| void RemoveLogo()                           | Removes company logo  |

**See also**

[Image interface](#)  
[iSpring SDK COM API](#)

### 1.3.6. iSpring.CorePlugin Interface

The **CorePlugin** Interface provides information about player core plugin.

| Properties |            |           |               |                      |
|------------|------------|-----------|---------------|----------------------|
| Property   | Value Type | Access    | Default value | Property description |
| Name       | string     | Read only |               | Core plugin name     |

| Methods       |                    |
|---------------|--------------------|
| Method        | Method description |
| void Delete() | Deletes the plugin |

**See also**

[CorePlugins Interface](#)  
[iSpring SDK COM API](#)

**1.3.7. iSpring.CorePlugins Interface**

The **CorePlugins** Interface represents a collection of core plugins. Core plugins are Flash modules which extend the standard functionality of the presentation playback core.

| Properties |            |           |               |   |
|------------|------------|-----------|---------------|---|
| Property   | Value Type | Access    | Default value | Property description                                    |
| Count      | int        | Read only | 0             | Returns a number of core plugins attached to the player |

| Methods   |  |
|---|--|
| Method  | Method description   |
| <a href="#">CorePlugin</a> Item(int index)                                      | Returns the <a href="#">CorePlugin</a> Interface representing a player core plugin with the given plugin index.<br><b>Note:</b> index is 1-based |
| <a href="#">CorePlugin</a> AddBuiltInPlugin ( <a href="#">BuiltInPluginType</a> | Adds built-in plugin. The plugin may be one of   |

|         |  |
|---------|--|
| plugin) | <a href="#">BuiltInPluginType</a> Enumeration. |
|---------|--|

**See also**

- [BuiltInPluginType Enumeration](#)
- [CorePlugin Interface](#)
- [PlayerSettings Interface](#)
- [iSpring SDK COM API](#)

**1.3.8. iSpring.FlashVideoFormat Enumeration**

**FlashVideoFormat** Enumeration defines various video formats supported by Adobe Flash player. Possible values of the FlashVideoFormat enumeration are:

- VIDEO\_FORMAT\_FLV** = 0,
- VIDEO\_FORMAT\_SWF** = 1

**See also**

- [VideoParameters Interface](#)
- [iSpring SDK COM API](#)

**1.3.9. iSpring.Image Interface**

The **Image** interface provides information about picture used as presenter photo or company logo.

| Properties |            |           |               |                      |
|------------|------------|-----------|---------------|----------------------|
| Property   | Value Type | Access    | Default value | Property description |
| Width      | int        | Read only |               | Returns image width  |
| Height     | int        | Read only |               | Returns image height |

| Methods   |   |
|---|---|
| Method  | Method description  |
| void Export(string path, int width, int height, int jpegQuality); | Exports the image to the file with specified path. File format depends on the file extension and can be one |

|  |   |
|--|---|
|  | <p>of followings:<br/> JPG<br/> PNG<br/> GIF.<br/> width and height parameters specify image size. These parameters can be equal to zero. The next cases are possible:<br/> width = 0, height = 0. Image will be saved with original size.<br/> width = 0, height &gt; 0. width will be calculated using height value and original aspect ratio.<br/> height = 0, width &gt; 1. height will be automatically calculated for keeping original aspect ratio for specified width.<br/> jpegQuality parameter is used only if JPG file format is specified.</p> |
|--|---|

**See also**

[iSpring SDK COM API](#)

**1.3.10. iSpring.ImageFileType Enumeration**

The **ImageFileType** Enumeration defines possible image file types accepted by the **SaveThumbnails** method of the [Slides Interface](#).

Possible values are:

**IFT\_WMF** = 0,  
**IFT\_PNG** = 1,  
**IFT\_JPG** = 2,  
**IFT\_BMP** = 3

**See also**

[Slides Interface](#)

[iSpring SDK COM API](#)

**1.3.11. iSpring.MediaSettings Interface**

The **MediaSettings** Interface represents media format and compression settings.

| Properties |            |        |               |                      |
|------------|------------|--------|---------------|----------------------|
| Property   | Value Type | Access | Default value | Property description |
|            |            |        |               |                      |

|                     |                                  |            |        |  |
|---------------------|----------------------------------|------------|--------|--|
| AudioBitrate        | <a href="#">AudioBitrateEnum</a> | Read/write | AB_128 | Audio bitrate used for audio compression.<br><b>Note:</b> Possible values are:<br>AB_ORIGINAL = 0,<br>AB_8 = 8,<br>AB_16 = 16,<br>AB_24 = 24,<br>AB_32 = 32,<br>AB_40 = 40,<br>AB_48 = 48,<br>AB_56 = 56,<br>AB_64 = 64,<br>AB_80 = 80,<br>AB_96 = 96,<br>AB_112 = 112,<br>AB_128 = 128,<br>AB_160 = 160,<br>AB_192 = 192,<br>AB_224 = 224,<br>AB_256 = 256,<br>AB_320 = 320 |
| ExportAudio         | bool                             | Read/write | true   | Indicates whether audio files are processed  |
| ExportEmbeddedFlash | bool                             | Read/write | true   | Indicates whether embedded Flash movies are processed  |
| ExportVideo         | bool                             | Read/write | true   | Indicates whether video files are processed  |
| ImageQuality        | int                              | Read/write | 75     | JPEG quality of raster images. Possible values are in range 0:100.<br><b>Note:</b> ImageQuality=0 indicates that the lossless (PNG)  |

|                     |      |            |               |   |
|---------------------|------|------------|---------------|---|
|                     |      |            |               | compression is applied to images. Values from 1 to 100 specify lossy JPEG compression ratio. Higher ImageQuality values result in better quality and larger size of the generated Flash movies.                                   |
| GifImageQuality     | int  | Read/write | =ImageQuality | JPEG quality of animated GIF images. Possible values are in range 0:100. See ImageQuality property for details. <b>Note:</b> until it is set explicitly, the JpegImageQuality property value equals to the ImageQuality property, |
| JpegImageQuality    | int  | Read/write | =ImageQuality | JPEG quality of photorealistic images. Possible values are in range 0:100. <b>Note:</b> until it is set explicitly, the JpegImageQuality property value equals to the ImageQuality property                                       |
| RasterizeChartsAndC | bool | Read/write | false         | Indicates whether graphs and charts are stored in raster form. Setting this option to true can improve complex  |

|                      |      |            |       |   |
|----------------------|------|------------|-------|---|
|                      |      |            |       | charts and graphs conversion quality.<br><b>Note:</b> the default value of this property is false.  |
| StoreAudioExternally | bool | Read/write | false | Indicates whether audio files are embedded into the generated slides or stored externally.<br><b>Note:</b> this option affects only if a presentation is converted into a solid Flash file  |
| StoreFlashExternally | bool | Read/write | false | Indicates whether Flash movies are embedded into the generated slides or stored externally.<br><b>Note:</b> some Flash movies don't work properly being embedded into another Flash file. Storing them separately from the presentation helps to resolve this problem in most cases |
| StoreVideoExternally | bool | Read/write | false | Indicates whether video files are embedded into the generated Flash presentation or saved separately into .FLV files.<br><b>Note:</b> due to <a href="#">Flash format limitations</a> embedded video files can contain as many as 16000 frames (about 11                            |

|              |     |            |     |  |
|--------------|-----|------------|-----|--|
|              |     |            |     | minutes of 24 fps video). This limitation is removed when videos are stored in separate FLV files. |
| VideoBitrate | int | Read/write | 400 | Video files compression bit rate. Possible values are from 50 to 4000 kbit/sec                     |

**See also**

[PresentationConverter Class](#)  
[Settings Interface](#)  
[iSpring SDK COM API](#)  
[Flash format limitations](#)

**1.3.12. iSpring.MouseClickAdvanceType Enumeration**

The **MouseClickAdvanceType** Enumeration defines the possible values of the [AdvanceType](#) property of the [NavigationSettings](#) Interface.

**MCAT\_NEXT\_SLIDE** = 0 - mouse click advances to the next presentation slide

**MCAT\_NEXT\_STEP** = 1 - mouse click advances to the next animation step

**See also**

[NavigationSettings Interface](#)  
[iSpring SDK COM API](#)

**1.3.13. iSpring.NavigationSettings Interface**

The **NavigationSettings** Interface represents mouse and keyboard navigation settings and allows the hyperlink target window customization.

| Properties       |            |            |               |                                      |
|------------------|------------|------------|---------------|--------------------------------------|
| Property         | Value Type | Access     | Default value | Property description                 |
| AdvanceOnMouseCl | bool       | Read/write | true          | Indicates whether the advance to the |

|                  |                                     |            |               |   |
|------------------|-------------------------------------|------------|---------------|---|
|                  |                                     |            |               | next animation step or slide on mouse click is enabled  |
| AdvanceType      | <a href="#">MouseClickedAdvance</a> | Read/write | MCAT_NEXT_STE | Specifies the way mouse click advances the presentation - to the next slide or to the next animation step |
| EnableHyperlinks | bool                                | Read/write | true          | Indicates whether hyperlinks are preserved during conversion  |
| HyperlinksTarget | string                              | Read/write | _self         | Specifies a browser window name where hyperlinks are opened   |
| KeyboardEnabled  | bool                                | Read/write | true          | Indicates whether keyboard navigation is enabled.   |

**See also**

- [MouseClickedAdvanceType Enumeration](#)
- [Settings Interface](#)
- [iSpring SDK COM API](#)

**1.3.14. iSpring.PasswordSettings Interface**

The **PasswordSettings** Interface provides password settings for protection of presentation from unauthorized viewing.

| Properties |            |            |               |                      |
|------------|------------|------------|---------------|----------------------|
| Property   | Value Type | Access     | Default value | Property description |
| Password   | string     | Read/Write |               | Password value.      |

**See also**

[ProtectionSettings Interface](#)  
[iSpring SDK COM API](#)

**1.3.15. iSpring.PlaybackSettings Interface**

The **PlaybackSettings** Interface represents presentation playback settings.

| Properties              |            |            |               |   |
|-------------------------|------------|------------|---------------|---|
| Property                | Value Type | Access     | Default value | Property description  |
| AutomaticSlideAdvance   | bool       | Read/write | false         | <b>This property is obsolete.</b> To control slide advance mode you should use <b>ChangeSlidesAutomatically</b> and <b>UseMinimalSlideDuration</b> properties instead |
| AutomaticStepAdvance    | bool       | Read/write | false         | Indicates whether the next animation step advance is performed automatically  |
| AutoStart               | bool       | Read/write | true          | Indicates whether the presentation starts playing automatically   |
| EnableAnimationEffects  | bool       | Read/write | true          | Indicates whether animation effects are enabled   |
| EnableTransitionEffects | bool       | Read/write | true          | Indicates whether transition effects are enabled  |
| LoopPresentation        | bool       | Read/write | false         | Indicates whether   |

|                           |                                |            |       |  |
|---------------------------|--------------------------------|------------|-------|--|
|                           |                                |            |       | the generated Flash presentation is played in a loop   |
| Player                    | <a href="#">PlayerSettings</a> | Read only  |       | Returns the <a href="#">PlayerSettings Interface</a> providing player customization facilities   |
| SlideAdvanceTime          | float                          | Read/write | 5     | Slide advance time   |
| StepAdvanceTime           | float                          | Read/write | 5     | Animation step advance time  |
| ChangeSlidesAutomatically | bool                           | Read/write | false | Indicates whether slides of the generated Flash presentation advance automatically regardless the presentation settings  |
| UseMinimalSlideDuration   | bool                           | Read/write | false | <p>If the property is set to true the <b>MinimalSlideDuration</b> property will define the minimal duration of each slide.</p> <p><b>Note:</b> If the property is set to false and PowerPoint slide transition advance time was not defined or equal to 0 seconds, minimal slide the duration will be automatically set to 1.0 second to</p> |

|                      |       |            |     |  |
|----------------------|-------|------------|-----|--|
|                      |       |            |     | prevent too fast slide advance.  |
| MinimalSlideDuration | float | Read/write | 5.0 | Minimal slide duration. Property is used if the <b>UseMinimalSlideDuration</b> parameter is set to true. |

**See also**

[Settings Interface](#)

[PlayerSettings Interface](#)

[iSpring SDK COM API](#)

**1.3.16. iSpring.PlayerSettings Interface**

The **PlayerSettings** Interface provides player customization facilities.

| Properties  |                             |           |               |   |
|-------------|-----------------------------|-----------|---------------|---|
| Property    | Value Type                  | Access    | Default value | Property description  |
| CorePlugins | <a href="#">CorePlugins</a> | Read only |               | Returns the <a href="#">CorePlugins Interface</a> representing player core plugins collection |

**See also**

[PlaybackSettings Interface](#)

[CorePlugins Interface](#)

[iSpring SDK COM API](#)

**1.3.17. iSpring.Plugins Interface**

The **Plugins** Interface provides an access to the Presentation Converter plugins. Implementing custom plugin interfaces you can replace some elements of the iSpring SDK standard functionality with your own.

| Properties |
|------------|
|------------|

| Property     | Value Type                   | Access    | Default value | Property description  |
|--------------|------------------------------|-----------|---------------|---|
| VideoPlugins | <a href="#">VideoPlugins</a> | Read only |               | Returns the <a href="#">VideoPlugins</a> Interface which allows customization of the video processing functionality |

**See also**

- [PresentationConverter Class](#)
- [VideoPlugins Interface](#)
- [iSpring SDK COM API](#)

**1.3.18. iSpring.Presentation Interface**

The **Presentation** Interface represents currently opened presentation. It exposes the following properties and methods:

| Properties      |                                 |            |               |  |
|-----------------|---------------------------------|------------|---------------|--|
| Property        | Value Type                      | Access     | Default value | Property description   |
| BackgroundAudio | <a href="#">BackgroundAudio</a> | Read only  |               | Returns a <a href="#">BackgroundAudio Interface</a> providing presentation background audio customization facilities |
| Height          | int                             | Read/Write |               | Slide width in pixels*   |
| Object          | object                          | Read only  |               | Returns a <a href="#">PowerPoint.Presentation</a> COM object which provides properties and methods for accessing the |

|                     |                            |            |  |   |
|---------------------|----------------------------|------------|--|---|
|                     |                            |            |  | presentation  |
| Title               | string                     | Read/Write |  | Presentation title  |
| SlideHeight         | int                        | Read/Write |  | Slide height in pixels  |
| SlideWidth          | int                        | Read/Write |  | Slide width in pixels   |
| Slides              | <a href="#">Slides</a>     | Read only  |  | Returns a <a href="#">Slides Interface</a> that represents a collection of presentation slides  |
| Width               | long                       | Read/Write |  | Slide height in pixels*   |
| XmlContentFile      | string                     | Read/Write |  | Specifies a file name, where the presentation XML description will be stored after conversion. If this property is an empty string, no XML description file will be generated |
| Presenters          | <a href="#">Presenters</a> | Read only  |  | Returns <a href="#">Presenters Interface</a> that represents a collection of presentation presenters  |
| HasDefaultPresenter | bool                       | Read only  |  | A Boolean value indicating whether the presentation has the default presenter.  |

|                  |                           |            |  |                                 |
|------------------|---------------------------|------------|--|---------------------------------|
| DefaultPresenter | <a href="#">Presenter</a> | Read/Write |  | Default presentation presenter. |
|------------------|---------------------------|------------|--|---------------------------------|

| Methods                       |                                   |
|-------------------------------|-----------------------------------|
| Method                        | Method description                |
| void Close()                  | Closes the presentation <b>**</b> |
| void RemoveDefaultPresenter() | Removes the default presenter     |

### Remarks

\* Width and Height properties are obsolete and are kept only for the compatibility with existing code using earlier versions of the SDK. Use **SlideWidth** and **SlideHeight** instead

\*\* The **ClosePresentation()** method of the [PresentationConverter](#) Class is obsolete and is kept only for the compatibility with the existing code using earlier versions of the SDK. New applications should use the Close() method of the **Presentation** Interface instead.

### See also

[BackgroundAudio Interface](#)  
[PresentationConverter Class](#)  
[Presenter Interface](#)  
[Presenters Interface](#)  
[Slides Interface](#)  
[iSpring SDK COM API](#)

## 1.3.19. PresentationConverter Class

### 1.3.19.1. PresentationConverter.CancelConversion() Method

The **CancelConversion** method stops the conversion process and optionally deletes the generated files.

### Syntax

```
void CancelConversion(bool deleteGeneratedFiles = true)
```

### Parameters

***deleteGeneratedFiles*** - a boolean value which indicates whether the generated files must be deleted. The default value of this parameter is true.

### Description

This method stops the presentation conversion process which was started using **GenerateFlash**, **GenerateSolidPresentation**, **GenerateCompoundPresentation**, and **GenerateStandaloneSlides** methods of the [PresentationConverter](#) Class. It is usually called in the conversion event handler.

### Return value

none.

### Example

#### C# Sample

The following sample creates a simple window with **Convert** and **Cancel** buttons.

When the user clicks the Convert button, the "**c:\test.ppt**" file is converted to the "**c:\test.swf**" Flash file. During conversion process, currently processed slide index is displayed on the form. When user clicks the Cancel button, the conversion process is aborted using the **CancelConversion** method.

```
using System;
using System.Windows.Forms;

using iSpring;

namespace GUITest
{
    public partial class Form1 : Form
    {
        private PresentationConverter m_converter;

        public Form1()
        {
            InitializeComponent();

            m_converter = new PresentationConverter();

            // initialize event handlers
            m_converter.OnIdle += new
            _ISpringEvents_OnIdleEventHandler(converter_OnIdle);
            m_converter.OnSlideProgressChanged += new
            _ISpringEvents_OnSlideProgressChangedEventHandler(converter_OnSlideProgressChanged);
        }
    }
}
```

```
        void converter_OnSlideProgressChanged(int slideIndex, int
totalSlides)
        {
            // display currently processed slide
            currentSlideLabel.Text = "Current slide: " + (slideIndex + 1) +
" / " + totalSlides;
            Application.DoEvents();
        }

        void converter_OnIdle()
        {
            // handle window events during conversion
            Application.DoEvents();
        }

        private void btnConvert_Click(object sender, EventArgs e)
        {
            btnConvert.Enabled = false;
            btnCancel.Enabled = false;
            try
            {
                m_converter.OpenPresentation("c:\\test.ppt");
                m_converter.GenerateSolidPresentation("c:\\test.swf", "",
null);
            }
            catch(Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
            btnCancel.Enabled = false;
            btnConvert.Enabled = true;
        }

        private void btnCancel_Click(object sender, EventArgs e)
        {
            btnCancel.Enabled = false;
            m_converter.CancelConversion(true);
            btnConvert.Enabled = true;
        }
    }
}
```

**See also**

[PresentationConverter Class](#)  
[iSpring SDK COM API](#)

### 1.3.19.2. LastErrorDescription Property

#### Description

The **LastErrorDescription** Property stores a brief description of the last occurred error.

**Type**

string (read only)

**Example**

The following samples illustrate how to get error description during presentation opening.

**VB Sample**

```
Sub iSpringSample()  
    On Error GoTo e  
  
    Dim pc As New PresentationConverter  
    pc.OpenPresentation "c:\presentation.ppt"  
  
    Exit Sub  
e:  
  
    MsgBox "Error:" & pc.LastErrorDescription  
End Sub
```

**C# sample**

```
using System;  
using iSpring;  
  
class iSpringSample  
{  
    static void Main(string[] args)  
    {  
        PresentationConverter pc = new PresentationConverter();  
        try  
        {  
            pc.OpenPresentation("D:\\problemSlide.ppt");  
        }  
        catch (Exception)  
        {  
            Console.WriteLine("Error: " + pc.LastErrorDescription);  
        }  
    }  
}
```

**See also**

[PresentationConverter Class](#)  
[iSpring SDK COM API](#)

**1.3.19.3. LicenseName Property**

## Description

The **LicenseName** Property stores the license name of the iSpring copy.

## Type

String (read/write)

## Remarks

**LicenseName** and [LicenseNumber](#) properties are used to pass iSpring SDK license details to the [PresentationConverter Object](#) and remove evaluation mode banner from generated Flash presentations.

The detailed information about purchase of iSpring SDK license can be found on [iSpring Purchase Page](#).

## Example

The following samples illustrate how to pass license details to the [PresentationConverter Object](#) and generate a solid Flash file without evaluation banner.

### VB Sample

```
Sub iSpringSample()  
    Dim pc As New iSpring.PresentationConverter  
  
    pc.LicenseName = "John Smith"  
    pc.LicenseNumber = "BVTCR-BYD21-ZPTDR-4BPQM-940M9-KN4XQ-KP"  
  
    pc.OpenPresentation "c:\test.ppt"  
  
    pc.GenerateSolidPresentation "c:\test.swf"  
End Sub
```

### C# Sample

```
using System;  
using iSpring;  
  
class iSpringSample  
{  
    static void Main(string[] args)  
    {  
        PresentationConverter pc = new PresentationConverter();  
  
        pc.LicenseName = "John Smith";  
        pc.LicenseNumber = "BVTCR-BYD21-ZPTDR-4BPQM-940M9-KN4XQ-KP";  
  
        pc.OpenPresentation("c:\\test.ppt");  
    }  
}
```

```
    pc.GenerateSolidPresentation("c:\\test.swf", "", null);  
  }  
}
```

## See Also

[LicenseNumber Property](#)  
[PresentationConverter Class](#)  
[iSpring SDK COM API](#)

### 1.3.19.4. LicenseNumber Property

#### Description

The **LicenseNumber** Property stores the license number of the iSpring copy.

#### Type

String (read/write)

#### Remarks

[LicenseName](#) and **LicenseNumber** properties are used to pass iSpring SDK license details to the [PresentationConverter Object](#) and remove evaluation mode banner from generated Flash presentations.

The detailed information about purchase of iSpring SDK license can be found on [iSpring Purchase Page](#).

#### Example

The following samples illustrate how to pass license details to the [PresentationConverter Object](#) and generate a solid Flash file without evaluation banner.

#### VB Sample

```
Sub iSpringSample()  
  Dim pc As New iSpring.PresentationConverter  
  
  pc.LicenseName = "John Smith"  
  pc.LicenseNumber = "BVTCTCR-BYD21-ZPTDR-4BPQM-940M9-KN4XQ-KP"  
  
  pc.OpenPresentation "c:\\test.ppt"  
  
  pc.GenerateSolidPresentation "c:\\test.swf"  
End Sub
```

#### C# Sample

```
using System;
using iSpring;

class iSpringSample
{
    static void Main(string[] args)
    {
        PresentationConverter pc = new PresentationConverter();

        pc.LicenseName = "John Smith";
        pc.LicenseNumber = "BVTCR-BYD21-ZPTDR-4BPQM-940M9-KN4XQ-KP";

        pc.OpenPresentation("c:\\test.ppt");
        pc.GenerateSolidPresentation("c:\\test.swf", "", null);
    }
}
```

### See Also

[LicenseName Property](#)  
[PresentationConverter Class](#)  
[iSpring SDK COM API](#)

### 1.3.19.5. PresentationOpened Property

#### Description

The **PresentationOpened** property indicates if the presentation has been opened and is accessible via **Presentation** property of the [PresentationConverter](#) object.

#### Type

Boolean (read only)

#### Example

The following examples illustrate how to check if the presentation was opened.

#### VB Sample

```
Sub iSpringSample()
    Dim pc As New iSpring.PresentationConverter

    pc.OpenPresentation "c:\test.ppt"

    If pc.PresentationOpened Then
        pc.GenerateSolidPresentation "c:\test.swf"
    End If
End Sub
```

#### C# Sample

```
using System;
using ISpring;

class iSpringSample
{
    static void Main(string[] args)
    {
        PresentationConverter pc = new PresentationConverter();

        pc.OpenPresentation("c:\\test.ppt");

        if (pc.PresentationOpened)
        {
            pc.GenerateSolidPresentation("c:\\test.swf", "", null);
        }
    }
}
```

**See also**

[PresentationConverter Class](#)  
[iSpring SDK COM API](#)

**1.3.19.6. Version Property****Description**

The **Version** property returns iSpring SDK version in the following format:

<major version>.<minor version>.<maintenance release>.<build number>

**Type**

string (read only)

**Remarks**

You can use the **Version** property to determine if iSpring SDK of the required version is installed and the version-specific functionality is available.

**Example**

The following examples display iSpring SDK version.

**VB Sample**

```
Private Sub iSpringSample()
    Dim pc As New iSpring.PresentationConverter

    MsgBox "iSpring SDK Version is " & pc.Version
```

End Sub

### C# Sample

```
using System;
using ISpring;

class iSpringSample
{
    static void Main(string[] args)
    {
        PresentationConverter pc = new PresentationConverter();
        Console.WriteLine("iSpring SDK version is " + pc.Version);
    }
}
```

### See also

[PresentationConverter Class](#)  
[iSpring SDK COM API](#)

### 1.3.20. iSpring.Presenter Interface

The Presenter Interface provides information about presentation presenter.

| Properties    |            |            |               |   |
|---------------|------------|------------|---------------|---|
| Property      | Value Type | Access     | Default value | Property description                      |
| Name          | string     | Read/Write | ""            | Presenter name                            |
| Title         | string     | Read/Write | ""            | Presenter title                           |
| WebSite       | string     | Read/Write | ""            | Presenter web site address                |
| Email         | string     | Read/Write | ""            | Presenter email address                   |
| BiographyText | string     | Read/Write | ""            | Brief information about the presenter     |
| HasPhoto      | bool       | Read only  | false         | Indicates whether the presenter has photo |

|                  |                         |           |       |   |
|------------------|-------------------------|-----------|-------|---|
| Photo            | <a href="#">Image</a>   | Read only |       | Returns the <a href="#">Image</a> interface providing information about presenter photo |
| HasCustomCompany | bool                    | Read only | false | Indicates whether the presenter has custom company                                      |
| CustomCompany    | <a href="#">Company</a> | Read only |       | Returns the Company interface providing information about presenter custom company.     |

| Methods                                      |   |
|--|---|
| Method                                       | Method description  |
| void Delete()                                | Deletes current presenter   |
| <a href="#">Image</a> LoadPhoto(string path) | Loads presenter photo from specified path and returns the <a href="#">Image</a> interface for this photo. |
| void RemovePhoto()                           | Removes presenter photo   |

**See also**[Image Interface](#)[Company Interface](#)[Presenters Interface](#)[Presentation Interface](#)[iSpring SDK COM API](#)**1.3.21. iSpring.Presenters Interface**

The Presenters Interface represents a collection of presentation presenters.

| Properties |            |        |               |                      |
|------------|------------|--------|---------------|----------------------|
| Property   | Value Type | Access | Default value | Property description |
|            |            |        |               |                      |

|                 |                           |            |  |   |
|-----------------|---------------------------|------------|--|---|
| Item(int index) | <a href="#">Presenter</a> | Read/Write |  | Returns the Presenter Interface representing a presentation presenter with the given index. <b>Note:</b> index is 1-based |
| Count           | int                       | Read only  |  | Returns a number of presenter within the presentation   |
| Company         | <a href="#">Company</a>   | Read/Write |  | Returns the Company interface providing information about presenters default company                                      |

| Methods                             |  |
|-------------------------------------|--|
| Method                              | Method description   |
| Presenter AddPresenter(string name) | Adds new presenter with specified name to the presentation presenters collection. Method returns Presenter interface for new added presenter |
| Void LoadFromPresentation()         | Loads information about presenters from the presentation file.   |
| Void SaveToPresentation()           | Saves information about presenters to the presentation file.   |

**See also**

[Presenter](#)  
[Company](#)  
[iSpring SDK COM API](#)

**1.3.22. iSpring.ProtectionSettings Interface**

The ProtectionSettings interface provides copyright protection facilities.

| Properties       |                                  |            |               |  |
|------------------|----------------------------------|------------|---------------|--|
| Property         | Value Type                       | Access     | Default value | Property description   |
| AllowDomain      | string                           | Read/Write |               | Domains that you wish to allow access to the presentation                            |
| RestrictedDomain | string                           | Read/Write |               | Domains that you wish to deny access to the presentation                             |
| PasswordSettings | <a href="#">PasswordSettings</a> | Read       |               | Returns the PasswordSettings interface providing information about password settings |
| Watermark        | <a href="#">Watermark</a>        | Read       |               | Returns the Watermark interface providing information about watermark settings       |

**See also**[PasswordSettings](#)[Watermark](#)[iSpring SDK COM API](#)**1.3.23. iSpring.Publish Interface**

The **PublishSettings** Interface represents presentation publish settings.

| Properties |            |            |               |   |
|------------|------------|------------|---------------|---|
| Property   | Value Type | Access     | Default value | Property description                      |
| Xml        | string     | Read/write |               | A string containing XML representation of |

|  |  |  |  |                      |
|--|--|--|--|----------------------|
|  |  |  |  | the publish settings |
|--|--|--|--|----------------------|

| Methods                    |                                      |
|----------------------------|--------------------------------------|
| Method                     | Method description                   |
| void Load(string fileName) | Loads publish settings from XML file |
| void Save(string fileName) | Saves publish settings to XML file   |

**See also**

[Skin Interface](#)

[iSpring SDK COM API](#)

**1.3.24. iSpring.Settings Interface**

This interface represents conversion settings of the PresentationConverter class. It provides the following properties and methods.

| Properties  |                                    |            |               |  |
|-------------|------------------------------------|------------|---------------|--|
| Property    | Value Type                         | Access     | Default value | Property description   |
| AllowDomain | string                             | Read/write | "*"           | A name of the domain which Flash files are allowed to access the ActionScript API of the generated Flash presentations.<br><b>Note:</b> the generated presentation passed the specified domain name to the <a href="#">System.security.allowDomain()</a> method. |
| Appearance  | <a href="#">AppearanceSettings</a> | Read only  |               | Returns an <a href="#">AppearanceSettings</a> Interface providing Flash presentation   |

|                 |                                    |            |       |   |
|-----------------|------------------------------------|------------|-------|---|
|                 |                                    |            |       | appearance customization facilities   |
| CpuLimit        | int                                | Read/write | 100   | Specifies an approximate CPU usage limit during conversion<br><b>Note:</b> integer values from 10 to 100 percent are accepted                                       |
| DisablePrinting | bool                               | Read/Write | false | Disables the "Print..." option in the Adobe Flash player context menu*  |
| GenerateExe     | bool                               | Read/Write | false | Indicates whether a standalone Flash player is attached to the generated Flash content allowing playback on windows system with no Flash player or addon installed. |
| LogFile         | string                             | Read/write | "     | The log file path. A log file contains the information about the presentation conversion progress.  |
| Media           | <a href="#">MediaSettings</a>      | Read only  |       | Returns a <a href="#">MediaSettings</a> Interface providing multimedia options and compression control facilities   |
| Navigation      | <a href="#">NavigationSettings</a> | Read only  |       | Returns a   |

|                  |                                  |            |   |   |
|------------------|----------------------------------|------------|---|---|
|                  |                                  |            |   | <a href="#">NavigationSettings</a><br>Interface providing the Flash presentation navigation settings  |
| Playback         | <a href="#">PlaybackSettings</a> | Read only  |   | Returns a <a href="#">PlaybackSettings</a> Interface providing Flash presentation playback control facilities   |
| RestrictedDomain | string                           | Read/write | " | The domain name (or names separated by semicolon) where the generated Flash presentation playback is allowed. This property prevents Flash presentation playback on unauthorized domain or local computer. If the RestrictedDomain property is an empty string (a default value), Flash presentations will be generated without any playback restrictions |
| SwfVersion       | int                              | Read/write | 7 | Adobe Flash player version required to play the generated Flash movie   |
| TemporaryFolder  | string                           | Read/write | " | A path to the folder where temporary files will be created  |

|                     |                                    |            |       |  |
|---------------------|------------------------------------|------------|-------|--|
|                     |                                    |            |       | during generatrion.<br><b>Note:</b> temporary files are automatically deleted at the end of the presentation conversion  |
| ZipOutput           | bool                               | Read/write | false | A Boolean value indicating whether the generated files are to be compressed  |
| PreserveCompletePre | bool                               | Read/write | true  | A boolean value indicating that complete presentation info should be included into presentation description file even through it is not used by skin. Complete presentation information is accessible via API if the property value is true. |
| Protection          | <a href="#">ProtectionSettings</a> | Read       |       | Returns the ProtectionSettings interface provideing copyrights and protection and facilities   |

| Methods                    |   |
|----------------------------|---|
| Method                     | Method description                              |
| void Load(string fileName) | Loads the conversion settings from the XML file |
| void Save(string fileName) | Saves the settings to the XML file              |

**Remarks**

\* The **DisablePrinting** property of the Settings Interface is obsolete. Use the `DisablePrinting` property of the [AppearanceSettings](#) interface.

**See also**

[AppearanceSettings Interface](#)

[MediaSettings Interface](#)

[NavigationSettings Interface](#)

[PlaybackSettings Interface](#)

[PresentationConverter Class](#)

[iSpring SDK COM API](#)

**1.3.25. iSpring.Skin Interface**

The **Skin** Interface provides skin information.

| Properties      |                                 |           |               |   |
|-----------------|---------------------------------|-----------|---------------|---|
| Property        | Value Type                      | Access    | Default value | Property description  |
| Alias           | string                          | Read only |               | Skin alias. Skin alias is used for compatibility with earlier versions of SDK. See the <a href="#">remarks</a> section for more details |
| Id              | string                          | Read only |               | Skin identifier. Used for skin identification.  |
| Name            | string                          | Read only |               | Skin name. User friendly skin name. This name can be used in a skin selection dialog.   |
| PublishSettings | <a href="#">PublishSettings</a> | Read only |               | Returns the <a href="#">PublishSettings</a> Interface which represents  |

|            |                              |           |  |   |
|------------|------------------------------|-----------|--|---|
|            |                              |           |  | presentation publish settings   |
| Settings   | <a href="#">SkinSettings</a> | Read only |  | Returns the <a href="#">SkinSettings</a> Interface which represents skin settings     |
| SwfVersion | int                          | Read only |  | A minimal Adobe Flash Player version required for the skin. Possible values are 7, 8. |

### Remarks

Name, Id and Alias properties of the Skin interface can be passed to the following Flash generation methods of the PresentationConverter object:

- **GenerateSolidPresentation()**
- **GenerateCompoundPresentation()**
- **GenerateFlash()**

It is not recommended to specify the same id, name or alias attributes for different skins.

### See also

[PublishSettings Interface](#)

[Skins Interface](#)

[SkinSettings Interface](#)

[iSpring SDK COM API](#)

### 1.3.26. iSpring.Skins Interface

The **Skins** Interface represents a collection of available player skins. It provides the information about skins and allows adding customized skins. The Skins Interface exposes the following properties and methods.

| Properties |            |        |               |                      |
|------------|------------|--------|---------------|----------------------|
| Property   | Value Type | Access | Default value | Property description |
|            |            |        |               |                      |

|       |     |           |   |
|-------|-----|-----------|---|
| Count | int | Read only | Returns number of skins in the collection |
|-------|-----|-----------|---|

| Methods                                     |  |
|---|--|
| Method                                      | Method description   |
| <a href="#">Skin</a> Item(object skinIndex) | <p>Returns the <a href="#">Skin</a> interface representing a skin with a given skin index. The <b>skinIndex</b> parameter can be one of the following types:</p> <ul style="list-style-type: none"> <li>an <b>integer</b> value specifying skin order number. Note: order number is 1-based</li> <li>a <b>string</b> specifying skin name, id or alias. Skin id, name and alias are specified in the <b>.skin XML files</b> in the <b>{iSpringSDK installation folder}\players</b> folder.</li> </ul> <p><b>Note:</b> This is a default method of the Skins interface so it can be omitted in some programming languages such as Visual Basic or C#. See the <a href="#">Samples</a> section for more details.</p> |
| void Load(string path)                      | <p>Loads additional player skin or skins from the specified path.</p> <p><b>Note:</b> if the path parameter is a <b>folder name</b>, it will be searched for <b>*.skin</b> files. If the path parameter is a <b>file name</b>, the specified skin file name will be loaded.</p>  |

## Samples

The following samples illustrate typical tasks which can be performed using the Skins interface.

### C# sample

```
PresentationConverter pc = new PresentationConverter();
Skins skins = pc.Skins;

// the following ways can be used to access individual skins of the
collection
Skin skin1 = skins[1];
Skin skin2 = skins.Item(2);
Skin nullSkin = skins["NullSkin"];
Skin crystalSkin = skins["Crystal"];

// load one skin file
```

```
skins.Load("c:\\my skins\\skin1.skin");

// load skin files using wildcard
skins.Load("c:\\all skins\\skin*.skin");

// load all .skin files in c:\\myskins folder
skins.Load("c:\\myskins")

// Enumerate the installed skins
foreach (Skin skin in skin)
{
    Console.WriteLine(
        "Skin \t" + "name: \"" + skin.Name + "\", id: \"" + skin.Id +
        "\"\n" +
        "\t" + "alias: \"" + skin.Alias + "\", Flash Player Version: " +
        skin.SwfVersion
    );
}
```

**VB.NET sample:**

```
Dim pc as new PresentationConverter()

' the following ways can be used to access individual skins of the Skins
collection
Dim skins as Skins = pc.Skins
Dim skin1 as Skin = skins(1)
Dim skin2 as Skin = skins(2)
Dim nullSkin as Skin = skins("NullSkin")
Dim advancedSkin as Skin = skins.Item("Advanced")

// load one skin file
skins.Load("c:\\my skins\\skin1.skin")

// load skin files using wildcard
skins.Load("c:\\all skins\\skin*.skin")

// load all .skin files in c:\\myskins folder
skins.Load("c:\\myskins")
```

**See also**[PresentationConverter Class](#)[Skin Interface](#)[iSpring SDK COM API](#)**1.3.27. iSpring.SkinSettings Interface**

The **SkinSettings** Interface represents skin settings.

| Properties |            |        |               |          |
|------------|------------|--------|---------------|----------|
| Property   | Value Type | Access | Default value | Property |
|            |            |        |               |          |

|     |        |            |  |   |
|-----|--------|------------|--|---|
|     |        |            |  | description   |
| Xml | string | Read/write |  | A string containing XML representation of the skin settings |

|                            |                                       |
|----------------------------|---------------------------------------|
| Methods                    |                                       |
| Method                     | Method description                    |
| void Load(string fileName) | Loads skin settings from the XML file |
| void Save(string fileName) | Saves skin settings to the XML file   |

**See also**

[Skin Interface](#)

[iSpring SDK COM API](#)

**1.3.28. iSpring.Slide Interface**

The **Slide Interface** provides slide information.

| Properties         |            |            |               |   |
|--------------------|------------|------------|---------------|---|
| Property           | Value Type | Access     | Default value | Property description  |
| BackgroundAudioFil | string     | Read/write | "             | Slide background audio file name  |
| Hidden             | bool       | Read only  |               | Indicates whether the slide is hidden   |
| Id                 | int        | Read only  |               | Slide id  |
| Index              | int        | Read only  |               | Slide index within the presentation<br><b>Note:</b> slide index is zero-based |

|              |                           |            |  |   |
|--------------|---------------------------|------------|--|---|
| Notes        | string                    | Read/write |  | Slide notes text                            |
| Text         | string                    | Read only  |  | Slide text                                  |
| Title        | string                    | Read/write |  | Slide title                                 |
| Hidden       | bool                      | Read       |  | Indicates whether the slide is hidden       |
| HasPresenter | bool                      | Read       |  | Indicates whether the slide has a presenter |
| Presenter    | <a href="#">Presenter</a> | Read/write |  | Presenter associated with the slide.        |

| Methods  |  |
|--|--|
| Method   | Method description   |
| void SaveImage(string fileName, int width, int height, int quality=75) | Saves slide thumbnail to the specified file<br><b>Note:</b> the following image file types are supported: <ul style="list-style-type: none"> <li>• JPG</li> <li>• PNG</li> <li>• BMP</li> <li>• WMF</li> </ul> |
| void RemovePresenter()   | Removes presenter from the slide   |

**See Also**

[Slides Interface](#)  
[iSpring SDK COM API](#)

**1.3.29. iSpring.Slides Interface**

The **Slides** Interface represents a collection of presentation slides.

| Properties |            |        |               |                      |
|------------|------------|--------|---------------|----------------------|
| Property   | Value Type | Access | Default value | Property description |
|            |            |        |               |                      |

|       |     |           |  |  |
|-------|-----|-----------|--|--|
| Count | int | Read only |  | Returns a number of slides in the presentation |
|-------|-----|-----------|--|--|

| Methods   |   |
|---|---|
| Method  | Method description  |
| <a href="#">Slide</a> Item(int index)   | Returns the <a href="#">Slide</a> Interface representing a particular slide in the collection<br><b>Note:</b> index is 1-based<br>It is a default method the Slides interface, so it can be omitted in some programming languages such as C# or VisualBasic |
| void SaveThumbnails(string destinationFolder, string filePrefix, <a href="#">ImageFileType</a> format, int width, int height, int quality = 75) | Saves slide thumbnails to the specified folder. The quality argument makes sense only for IFT_JPG file type.  |

### Samples

The following C# sample illustrates Slides interface usage.

```
import iSpring;
...
PresentationConverter pc = new PresentationConverter();
pc.OpenPresentation("C:\\presentation.ppt");
Slides slides = pc.Presentation.Slides;
Slide firstSlide = slides[1];
Slide lastSlide = slides.Item(slides.Count);
slides.SaveThumbnails("C:\\thumbnails", "thumb", ImageFileType.IFT_JPG,
120, 90, 75);
```

### See Also

- [ImageFileType Enumeration](#)
- [Presentation Interface](#)
- [Slide Interface](#)
- [iSpring SDK COM API](#)

### 1.3.30. iSpring.VideoConverter Interface

The VideoConverter interface provides facilities for video files conversion.

| Methods  |   |
|--|---|
| Method   | Method description  |
| <a href="#">VideoConversionResult</a> ConvertVideo(string sourceFilePath, string destinationFilePath, VideoParameters VideoParameters) | Converts the source video file to the new video file with parameters specified in <a href="#">VideoParameters</a> |

**See also**

[VideoConversionResult](#)

[VideoParameters](#)

[iSpring SDK COM API](#)

### 1.3.31. iSpring.VideoConversionResult Enumeration

**VideoConversionResult** Enumeration defines the possible results of the video file conversion process. Possible values are:

**VCR\_SUCCESS** = 0, // video was converted successfully

**VCR\_USE\_DEFAULT\_CONVERTER** = 1, // this video should be processed by default converter

**VCR\_SKIP\_VIDEO** = 2, // the video object should be skipped

**See also**

[VideoConverter Interface](#)

[iSpring SDK COM API](#)

### 1.3.32. iSpring.VideoObjectInfo Interface

The VideoObjectInfo interface stores information about video object.

| Properties         |            |           |               |   |
|--------------------|------------|-----------|---------------|---|
| Property           | Value Type | Access    | Default value | Property description  |
| AnimationStepIndex | int        | Read only |               | Returns index of animation when the video object will be shown on the slide. Note. If there is no |

|                       |                                     |           |  |   |
|-----------------------|-------------------------------------|-----------|--|---|
|                       |                                     |           |  | animation step for the video object value is 0.   |
| DelayTime             | float                               | Read only |  | Returns time in seconds when the video object will be shown on the slide                                |
| DestinationSlideIndex | int                                 | Read      |  | Returns destination slide index contained the video object. Note. Index is 1- based                     |
| Height                | float                               | Read only |  | Returns height of the video object.   |
| Id                    | int                                 | Read only |  | Returns the identifier of the video object on the slide   |
| Left                  | float                               | Read only |  | Returns the distance in pixels between the left edge of the video object and the left edge of the slide |
| Path                  | string                              | Read only |  | Returns the path to the video object file   |
| PlaybackMode          | <a href="#">VideoObjectPlayback</a> | Read only |  | Returns the playback mode. Possible values are: VOPM_AUTOMATIC = 0, VOPM_MANUAL                         |
| SlideId               | int                                 | Read only |  | Returns an identifier of the slide containing the video object  |

|                  |                                 |           |  |   |
|------------------|---------------------------------|-----------|--|---|
| SourceSlideIndex | int                             | Read only |  | Returns slide index of the presentation contained the video object.<br><b>Note:</b> Index is 1-based.             |
| Top              | float                           | Read only |  | Returns the distance in pixels between the top edge of the video object and the top edge of the slide             |
| Type             | <a href="#">VideoObjectType</a> | Read only |  | Returns the format of the flash video object. Possible values are:<br>VOT_SLIDE_VIDEO = 0,<br>VOT_PRESENTER_VIDEO |
| Width            | float                           | Read only |  | Returns width in pixels of the video object   |

**See also**[VideoObjectPlaybackMode](#)[VideoObjectType](#)[iSpring SDK COM API](#)**1.3.33. iSpring.VideoObjectPlaybackMode Enumeration**

**VideoObjectPlaybackMode** Enumeration defines the possible modes of the slide video object playback. Possible values are:

**VOPM\_AUTOMATIC** = 0,

**VOPM\_MANUAL** = 1

**See also**[VideoObjectProcessor Interface](#)[iSpring SDK COM API](#)

### 1.3.34. iSpring.VideoObjectProcessingResult Enumeration

**VideoObjectProcessingResult** Enumeration defines the possible results of the slide video objects processing . Possible values are:

**VOPR\_PROCEED\_WITH\_THIS\_OBJECT** = 0, // proceed this video object  
**VOPR\_SKIP\_THIS\_OBJECT** = 1 // drop this object out of the presentation

See also

[VideoObjectProcessor Interface](#)  
[iSpring SDK COM API](#)

### 1.3.35. iSpring.VideoObjectType Enumeration

The **VideoObjectType** Enumeration defines the possible video object types which can be contained in the presentation. Possible values are:

**VOT\_SLIDE\_VIDEO** = 0,  
**VOT\_PRESENTER\_VIDEO** = 1

See also

[VideoObjectInfo Interface](#)  
[iSpring SDK COM API](#)

### 1.3.36. iSpring.VideoParameters Interface

The VideoParameters Interface stores video file conversion parameters.

| Properties |                                  |        |               |  |
|------------|----------------------------------|--------|---------------|--|
| Property   | Value Type                       | Access | Default value | Property description   |
| format     | <a href="#">FlashVideoFormat</a> | Read   |               | Returns the format of the flash video. Possible values are: VIDEO_FORMAT_FLV = 0, VIDEO_FORMAT_SWF |
| FrameRate  | float                            | Read   |               | Returns the framerate of the video.  |

|        |     |      |  |                                 |
|--------|-----|------|--|---------------------------------|
| Width  | int | Read |  | Returns the width of the video  |
| Height | int | Read |  | Returns the height of the video |

**See also**

[FlashVideoFormat](#)  
[iSpring SDK COM API](#)

**1.3.37. iSpring.VideoPlugins Interface**

The **VideoPlugins** Interface allows replacing standard video processing functionality with the custom one.

| Properties                 |                                      |            |               |  |
|----------------------------|--------------------------------------|------------|---------------|--|
| Property                   | Value Type                           | Access     | Default value | Property description   |
| CustomVideoConverter       | <a href="#">VideoConverter</a>       | Read/Write |               | Defines an object implementing the <a href="#">VideoConverter</a> interface which will replace the standard video conversion module of iSpring SDK.      |
| CustomVideoObjectProcessor | <a href="#">VideoObjectProcessor</a> | Read/Write |               | Defines an object implementing the <a href="#">VideoObjectProcessor</a> interface which will replace the standard video processing module of iSpring SDK |
| HasCustomVideoConverter    | bool                                 | Read only  | false         | Indicates whether the standard video converter is replaced with the custom one   |

|                   |      |           |       |   |
|-------------------|------|-----------|-------|---|
| HasCustomVideoObj | bool | Read only | false | Indicates whether the standard video object processor is replaced with the custom one |
|-------------------|------|-----------|-------|---|

| Methods                                 |   |
|---|---|
| Methods                                 | Methods description   |
| void RemoveCustomVideoConverter()       | Removes a custom video converter. When this method is invoked the built-in video conversion module of iSpring SDK will be used.               |
| void RemoveCustomVideoObjectProcessor() | Removes a custom video object processor. When this method is invoked the built-in video object processing module of iSpring SDK will be used. |

**See also**

- [VideoConverter](#)
- [VideoObjectProcessor Interface](#)
- [iSpring SDK COM API](#)

**1.3.38. iSpring.Watermark Interface**

The Watermark Interface provides facilities for adding custom watermarks to the generated Flash content

| Properties |                                   |            |               |  |
|------------|-----------------------------------|------------|---------------|--|
| Property   | Value Type                        | Access     | Default value | Property description   |
| Alignment  | <a href="#">WatermarkAlignmen</a> | Read/Write | WA_LEFT_TOP   | Alignment is used for adjusting of watermark position within the slide.<br><b>Note:</b> possible values are:<br>WA_LEFT_TOP = 1,<br>WA_CENTER_TOP = 2, |

|              |        |            |       |   |
|--------------|--------|------------|-------|---|
|              |        |            |       | WA_RIGHT_TOP<br>= 3,<br>WA_LEFT_MIDDLE<br>= 4,<br>WA_CENTER_MIDDLE<br>= 5,<br>WA_RIGHT_MIDDLE<br>= 6,<br>WA_LEFT_BOTTOM<br>= 7,<br>WA_CENTER_BOTTOM<br>= 8,<br>WA_RIGHT_BOTTOM<br>= 9 |
| ClickURL     | string | Read/Write |       | Web page address that will be opened on mouse click   |
| Enabled      | bool   | Read/Write | false | Indicate whether the watermark is enabled   |
| OffsetX      | int    | Read/Write | 0     | Horizontal offset of the watermark  |
| OffsetY      | bool   | Read/Write | 0     | Vertical offset of the watermark  |
| Path         | string | Read/Write |       | Path to bitmap or SWF file that will be used as watermark   |
| Target       | string | Read/Write | _self | Browser target window where web page will be opened   |
| Transparency | float  | Read/Write | 0.0   | Specifies watermark transparency. Possible values are in range 0.0 – 1.0. Note: transparency = 0 indicates that   |

|  |  |  |  |   |
|--|--|--|--|---|
|  |  |  |  | watermark will be fully opaque, transparency = 1.0 indicates that watermark will be fully transparent |
|--|--|--|--|---|

**See also**

[WatermarkAlignment](#)  
[iSpring SDK COM API](#)

### 1.3.39. iSpring.WatermarkAlignment Enumeration

**WatermarkAlignment** Enumeration defines the possible modes of the watermark position alignment. Possible values are:

**WA\_LEFT\_TOP** = 1,  
**WA\_CENTER\_TOP** = 2,  
**WA\_RIGHT\_TOP** = 3,  
**WA\_LEFT\_MIDDLE** = 4,  
**WA\_CENTER\_MIDDLE** = 5,  
**WA\_RIGHT\_MIDDLE** = 6,  
**WA\_LEFT\_BOTTOM** = 7,  
**WA\_CENTER\_BOTTOM** = 8,  
**WA\_RIGHT\_BOTTOM** = 9

**See also**

[Watermark Interface](#)  
[iSpring SDK COM API](#)

## 1.4. iSpring SDK command line interface

iSpring SDK contains a command line application **iSpringSDK.exe** which is based on [iSpring SDK COM API](#). This application provides a command line interface to the iSpring SDK functionality and can be used to perform batch conversion from a command line or shell scripts and other automation tools.

### Table of contents

- [iSpring SDK command line syntax](#)
- [Presentation conversion options](#)
  - [Compression options](#)

- [Animation and playback options](#)
- [Appearance options](#)
- [Output control options](#)
- [Multimedia control options](#)
- [Navigation options](#)
- [Licensing options](#)
- [Security options](#)
- [Miscellaneous options](#)
- [Thumbnail generation options](#)

### iSpring SDK command line syntax

The general iSpring command line application syntax is as follows.

```
ispringsdk.exe <command> [command options]
```

If the command line application is started without any parameters, it will print the list of supported commands and options to the standard output.

The **command** parameter specifies an action the command application will perform. iSpring SDK commands can have short alias which simplify and accelerate command line application usage. The following commands are currently supported.

| Commands       |               |   |
|----------------|---------------|---|
| Command        | Short synonym | Command description   |
| make-solid-swf | s             | <p>Performs batch conversion of PowerPoint presentations to solid Flash files. The syntax is as follows.</p> <pre>ispringsdk.exe make-solid-swf [conversion-options] &lt;swf&gt; ...] or ispringsdk.exe s [conversion-options]</pre> <p>Examples:</p> <pre>ispringsdk.exe s "c:\presentations\k c:\flash\business-plan.swf ispringsdk.exe s presentation1.ppt p presentation2.swf</pre> |
| make-slides    | l             | <p>Performs batch conversion of PowerPoint presentations to standalone Flash slides if a player <a href="#">skin</a> is not set, or compound Flash</p>  |

|                 |   |  |
|-----------------|---|--|
|                 |   | <p>presentation if player <a href="#">skin</a> is specified. The syntax is as follows.</p> <pre>ispringsdk.exe <b>make-slides</b> [<a href="#">conversion-options</a>] [&lt;ppt&gt; &lt;output-folder&gt; ...] or ispringsdk.exe <b>l</b> [<a href="#">conversion-options</a>] &lt;output-folder&gt; ...]</pre> <p>Examples:</p> <pre>ispringsdk.exe l "c:\my presentation files\business-plan" ispringsdk.exe l presentation1.ppt p presentation2</pre> |
| save-thumbnails | t | <p>Exports presentation slides to images of the specified format and resolution. The syntax is as follows.</p> <pre>ispringsdk.exe <b>save-thumbnails</b> [<a href="#">thumbnails-options</a>] &lt;output-folder&gt; or ispringsdk.exe <b>t</b> [<a href="#">thumbnails-options</a>]</pre> <p>Examples:</p> <pre>ispringsdk.exe s "c:\presentations\b c:\flash\business-plan.swf ispringsdk.exe s presentation1.ppt p presentation2.swf</pre>            |
| activate        | a | <p>Performs an activation of iSpring SDK. The syntax is as follows.</p> <pre>ispringsdk.exe <b>activate</b> &lt;license name&gt; or ispringsdk.exe <b>a</b> &lt;license name&gt; &lt;lic</pre> <p>Examples:</p> <pre>ispringsdk.exe a "John Lewis" "BYN6S-ZK10N-BHHDG-Q4658-L9PEE-7CW9F</pre>  |
| deactivate      | d | <p>Performs a deactivation of iSpring SDK. The syntax is as follows.</p> <pre>ispringsdk.exe <b>deactivate</b> &lt;license name&gt; or ispringsdk.exe <b>d</b> &lt;license name&gt; &lt;lic</pre>  |

Examples:

```
ispringsdk.exe d "John Lewis"
"BYN6S-ZK10N-BHHDG-Q4658-L9PEE-7CW9"
```

## Presentation conversion options

Presentation conversion options affect the behavior of the **make-solid-swf** and **make-slides** commands. These options can be separated into the following categories.

- [Compression options](#)
- [Animation and playback options](#)
- [Appearance options](#)
- [Output control options](#)
- [Multimedia control options](#)
- [Navigation options](#)
- [Licensing options](#)
- [Security options](#)
- [Miscellaneous options](#)

## Compression options

Compression options specify the multimedia objects compression and allow controlling the size of the generated Flash content.

| Key                             | Short key | Default value | Description  |
|---------------------------------|-----------|---------------|--|
| --audio-bitrate <bitrate>       | -ab       | 128           | Specifies audio files bitrate (kbit/sec). Possible values are: 8, 16, 24, 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256, 320 and "original" |
| --video-bitrate <bitrate>       | -vb       | 400           | Specifies video files bitrate. Possible values are between 50 and 4000   |
| --image-quality <image quality> | -iq       | 75            | Specifies a compression ratio applied to non-photorealistic images such as charts, graphs, diagrams and clip art objects.                              |

|                                     |      |     |  |
|-------------------------------------|------|-----|--|
|                                     |      |     | <p><b>Note:</b> possible values are <b>integers from 1 to 100</b> specifying lossy JPEG compression quality, or an "<b>original</b>" value meaning that lossless PNG compression is applied.<br/>Higher JPEG compression quality values result in higher image quality, the lower values result in smaller Flash content size.</p>   |
| --jpeg-image-quality <jpeg quality> | -jiq | 75* | <p>Specifies a compression ratio applied to photorealistic images such as photos and complex pictures.<br/><b>Note:</b> possible values are <b>integers from 1 to 100</b> specifying lossy JPEG compression quality, or an "<b>original</b>" value meaning that lossless PNG compression is applied.<br/>Higher JPEG compression quality values result in higher image quality, the lower values result in smaller Flash content size.</p> |
| --gif-image-quality <gif quality>   | -giq | 75* | <p>Specifies a compression ratio applied to animated GIF images.<br/><b>Note:</b> possible values are <b>integers from 1 to 100</b> specifying lossy JPEG compression quality, or an "<b>original</b>" value meaning that lossless PNG compression is applied.<br/>Higher JPEG</p>   |

|  |  |  |   |
|--|--|--|---|
|  |  |  | compression quality values result in higher image quality, the lower values result in smaller Flash content size. |
|--|--|--|---|

\*When **--jpeg-image-quality** or **--gif-image-quality** option is not specified explicitly, its value is equal to the **--image-quality** option.

In most cases the highest image quality to Flash content size ratio is provided by setting an image quality to the "original" value while jpeg and gif image quality values are between 65 and 85.

### Animation and playback options

Animation options control the way animations and t

| Key                                     | Short key  | Default value | Description   |
|---|------------|---------------|---|
| <b>--shape-animations</b><br><on/off>   | <b>-a</b>  | on            | Indicates whether animation effects are enabled   |
| <b>--transition-effects</b><br><on/off> | <b>-t</b>  | on            | Indicates whether transition effects are enabled  |
| <b>--loop-presentation</b><br><on/off>  | <b>-lp</b> | off           | Indicates whether the presentation playback is looped   |
| <b>--auto-step-advance</b><br><on/off>  | <b>-sa</b> | off           | Indicates whether slide animation steps are played automatically  |
| <b>--step-advance-time</b><br><time>    | <b>-st</b> | 5             | If the <b>--auto-step-option</b> option is set to on, the time parameter specifies step advance time. The possible values for this option are >0 and <= 100 seconds |
| <b>--framerate</b> <frame rate>         | <b>-f</b>  | 24            | Specifies Flash   |

|                                     |      |     |  |
|-------------------------------------|------|-----|--|
|                                     |      |     | presentation frame rate (number of frames per second). The bigger framerate - the smoother animations. |
| --auto-start <on/off>               |      | on  | Indicates whether presentation playback starts automatically   |
| --preserve-complete-info <on/off>   | -i   | on  | Indicates whether complete presentation info is to be available via ActionScript API                   |
| --export-slide-background <on/off>  | -eb  | on  | Specifies whether the slide background will be exported  |
| --change-slides-automatica <on/off> | -csa | off | Indicate whether the automatic slides change is forced.  |
| --minimal-slide-duration <seconds>  | -msd |     | Specifies the minimal slide duration. The possible values for this option are > 0 and <= 1000          |

### Appearance options

Appearance options control Flash presentation appearance.

| Key                     | Short key | Default value | Description                        |
|-------------------------|-----------|---------------|------------------------------------|
| --width <slide width>   | -w        |               | Specifies slide width (in pixels)  |
| --height <slide height> | -h        |               | Specifies slide height (in pixels) |
| --scale <scale>         | -s        |               | Specifies slide scale (in percent) |

|                                   |     |         |   |
|-----------------------------------|-----|---------|---|
|                                   |     |         | <b>Note:</b> this option must not be used with --width and --height options   |
| --screen-size <width><br><height> | -ss |         | If specified, optimizes raster images to the given screen resolution. This option allows to enhance raster images quality when the presentation is viewed full screen   |
| --image-smooth <on/off>           | -is | on      | Indicates whether smoothing is applied to raster images   |
| --bgcolor <#RRGGBB>               | -b  | #FFFFFF | Specifies flash movie background color in #RRGGBB format  |
| --fullscreen <on/off>             | -fs | off     | Indicates whether the presentation is playing full screen.<br><b>Note:</b> there are some <a href="#">full-screen mode restrictions</a> for Flash files running in browser window: <ul style="list-style-type: none"> <li>• Adobe Flash Player 9.0.28.0 or later is required</li> <li>• allowFullScreen tag parameter must be included to the &lt;object&gt; and &lt;embed&gt; tags of the HTML code.</li> <li>• Users cannot enter text in text input fields while in full-screen mode.</li> </ul> |
| --skin <skin name>                |     |         | The <b>skin name</b> parameter can be one of the following types:   |

|  |  |  |  |
|--|--|--|--|
|  |  |  | <ul style="list-style-type: none"> <li>• an <b>integer</b> value specifying skin order number. Note: order number is 1-based</li> <li>• a <b>string</b> specifying skin name, id or alias. Skin id, name and alias are specified in the <b>.skin</b> XML files in the <b>{iSpringSDK installation folder}\players</b> folder.</li> </ul> |
|--|--|--|--|

### Output control options

Output control options affect the generated content format or properties

| Key                       | Short key | Default value | Description  |
|---------------------------|-----------|---------------|--|
| --zip                     | -z        |               | Creates a .ZIP archive with the generated Flash content  |
| --xml <file name>         | -x        |               | Creates an XML file with the XML description of the generated Flash content  |
| --flash-version <version> | -fv       | 7             | Specifies an Adobe Flash Player version required to play the generated Flash content.<br><b>Notes:</b> possible values are 7, 8, and 9<br>This option guarantees that the generated Flash content will require the Flash player version no less than the specified one |
| --menu <on/off>           | -m        | on            | Enables or disables a context menu of the  |

|                     |    |    |  |
|---------------------|----|----|--|
|                     |    |    | Adobe Flash Player   |
| --printing <on/off> | -p | on | Enables or disables a "Print..." option in a context menu of the Adobe Flash Player  |
| --exe               |    |    | Wraps the generated Flash content into windows executable module allowing its playback in Windows-compatible system without Adobe Flash player installed |

### Multimedia control options

Multimedia control options affect the multimedia content conversion.

| Key                       | Short key | Default value | Description  |
|---------------------------|-----------|---------------|--|
| --export-audio <on/off>   | -ea       | on            | Indicates whether audio content of the original presentation is included into the generated Flash content          |
| --export-video <on/off>   | -ev       | on            | Indicates whether video of the original presentation is included into the generated Flash content                  |
| --export-flash <on/off>   | -ef       | on            | Indicates whether embedded Flash objects of the original presentation is included into the generated Flash content |
| --external-audio <on/off> | -xa       | off           | Indicates whether audio content is stored separately from the  |

|                                       |     |     |  |
|---------------------------------------|-----|-----|--|
|                                       |     |     | <p>generated Flash content.<br/> <b>Note:</b> this option affects only if a presentation is converted into a solid Flash file</p>  |
| --external-video <on/off>             | -xv | off | <p>Indicates whether video files are embedded into the generated Flash presentation or saved separately into .FLV files.<br/> <b>Note:</b> due to <a href="#">Flash format limitations</a> embedded video files can contain as many as 16000 frames (about 11 minutes of 24 fps video). This limitation is removed when videos are stored in separate FLV files.</p> |
| --external-flash <on/off>             | -xf | off | <p>Indicates whether Flash movies are embedded into the generated slides or stored externally.<br/> <b>Note:</b> some Flash movies don't work properly being embedded into another Flash file. Storing them separately from the presentation helps to resolve this problem in most cases.</p>  |
| --advanced-smart-art-process <on/off> |     | off | <p>Enables or disables advanced processing of Office 2007 Smart Art objects. Turning on this option can improve quality of Smart Art objects conversion. Being turned on, this option opens PowerPoint application window and uses Windows clipboard during conversion.</p>  |

## Navigation options

| Key                                    | Short key | Default value | Description   |
|--|-----------|---------------|---|
| --enable-hyperlinks<br><on/off>        | -hl       | on            | Indicates whether hyperlinks within a presentation are preserved  |
| --hyperlinks-target<br><target>        | -ht       | _self         | Specifies a browser window name where hyperlinks to external resources are opened                         |
| --advance-on-click<br><on/off>         | -c        | on            | Specifies whether slide or step advance on mouse click is enabled   |
| --click-advance-type<br><slide/step>   | -ct       | step          | Specifies the way mouse click advances the presentation - to the next slide or to the next animation step |
| --enable-keyboard-navigati<br><on/off> | -k        | on            | Indicates whether keyboard navigation is enabled  |

## Licensing options

| Key                              | Short key | Default value | Description   |
|----------------------------------|-----------|---------------|---|
| --license <name> <serial number> | -l        |               | Specifies registration name and serial number of the iSpring SDK license.<br>Note: when this option is omitted or incorrect registration name or serial number is specified, each slide of the generated Flash content will contain a watermark |

## Security options

| Key                                  | Short key | Default value | Description  |
|--------------------------------------|-----------|---------------|--|
| --allowed-domain<br><name>           | -ad       | *             | Specifies the name of the domain which Flash files are allowed to access the ActionScript API of the generated Flash content.<br><b>Note:</b> the generated presentation passed the specified domain name to the <a href="#">System.security.allowDomain()</a> method.   |
| --restricted-domain<br><name>        | -rd       |               | Specifies the domain name (or names separated by semicolon) where the generated Flash presentation playback is allowed. This property prevents Flash presentation playback on unauthorized domain or local computer. If this option is omitted, the generated Flash content is generated without any playback restrictions |
| --watermark <file-path>              | -wm       |               | Path to bitmap or SWF file that will be used as watermark  |
| --watermark-alignment<br><alignment> | -wa       | left-top      | Set watermark alignment within the slide. Possible <alignment> values are: left-top, center-top, right-top, left-middle, center-middle, right-middle, left-bottom, center-bottom, right-bottom   |

|  |       |       |  |
|--|-------|-------|--|
| --watermark-offset-x<br><offset x>         | -wx   | 0     | Set horizontal offset of watermark from the alignment point.   |
| --watermark-offset-y<br><offset y>         | -wy   | 0     | Set vertical offset of watermark from the alignment point.   |
| --watermark-transparency<br><transparency> | -wt   | 0.0   | Specifies watermark transparency. The possible values are in range from 0.0 to 1.0<br>Transparency = 0 indicates that the watermark is fully opaque while the transparency = 1 indicates that the watermark is fully transparent |
| --watermark-url <url>                      | -wu   |       | Specifies web page address to be opened on mouse click   |
| --watermark-target<br><target>             | -wtg  | _self | Specifies browser window name where the url specified with the <b>--watermark-url</b> option is to be opened   |
| --password <password>                      | -pass |       | Protects the generated Flash presentation from unauthorized viewing with the specified password  |

### Miscellaneous options

| Key               | Short key | Default value | Description  |
|-------------------|-----------|---------------|--|
| --cpu <cpu usage> |           | 100           | Specifies maximum CPU usage allowed for the presentation conversion. |

|                                  |       |  |   |
|----------------------------------|-------|--|---|
|                                  |       |  | <b>Note:</b> possible values are from 10 to 100   |
| --log <log file path>            |       |  | Specifies a log file name where presentation conversion events are stored   |
| --temp <temporary folder>        |       |  | Overrides a temporary files folder where temporary files are created during conversion  |
| --range <slide indices>          | -r    |  | Specifies slide indices to be converted to Flash format.<br><b>Example:</b><br><code>ispringsdk.exe s -r 1,3,8-15</code>  |
| --settings-file <file path>      | -sf   |  | Loads presentation conversion settings from the specified XML file.<br><b>Note:</b> The settings XML file can be created with iSpring SDK Code Builder or via iSpring SDK COM API using the <a href="#">Save()</a> method of the <a href="#">Settings Interface</a> . |
| --verbose                        | -v    |  | Outputs conversion progress to console  |
| --add-built-in-plugins <plugins> | -plug |  | Adds one or more player built-in plugins (comma-separated). The following plugin types are currently supported:<br><b>js</b> - <a href="#">Java script connection module</a> <b>as3</b> - <a href="#">ActionScript 3 connection module</a>                            |

## Thumbnails generation options

Thumbnail generation options affect the save-thumbnails command behavior

| Key                                   | Short key | Default value | Description   |
|---------------------------------------|-----------|---------------|---|
| --width <width>                       | -w        |               | Specifies thumbnail width   |
| --height <height>                     | -h        |               | Specifies thumbnails height   |
| --format<br><jpg/png/bmp/gif>         | -f        |               | Specifies thumbnails file format  |
| --prefix <thumbnail file name prefix> | -p        | thumb         | Specifies thumbnail file name prefix. Thumbnail names are generated according to the following scheme: <prefix><slide index>.<thumbnail format> |
| --quality <quality>                   | -q        | 75            | Specifies JPEG image quality for jpeg thumbnails. Possible values are in range 1-100  |
| --range <slide indices>               | -r        |               | Specifies slide indices which thumbnails will be generated for. Example:<br><code>ispringsdk.exe t -r 1,3,8-15</code>                           |

## 1.5. ActionScript API References

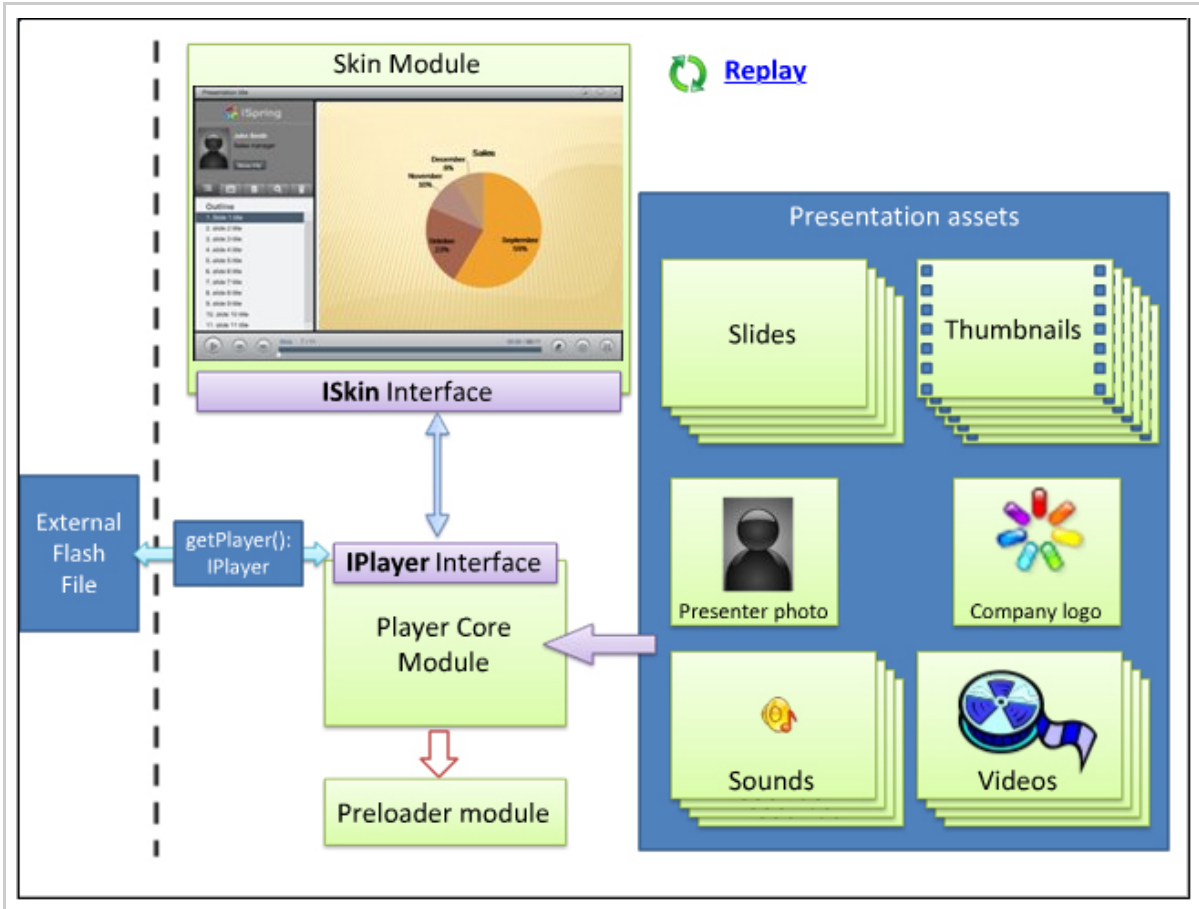
### 1.5.1. iSpring Presentation structure

Each iSpring generated Flash presentation consists of the following parts.

- [Presentation assets](#)
- [Player core module](#)

- [Skin module](#)

iSpring presentation structure is shown in the following diagram.



### 1.5.1.1. Presentation assets

Presentation assets are various multimedia resources containing within the presentation. When a presentation is a solid Flash movie file, all resources are stored within this file optimized for progressive downloading from the Internet. When a presentation is converted to a set of Flash files with player, each resource is stored in a separate file.

### 1.5.1.2. Player core module

Player core is a Flash module which loads presentation assets and controls slide animations

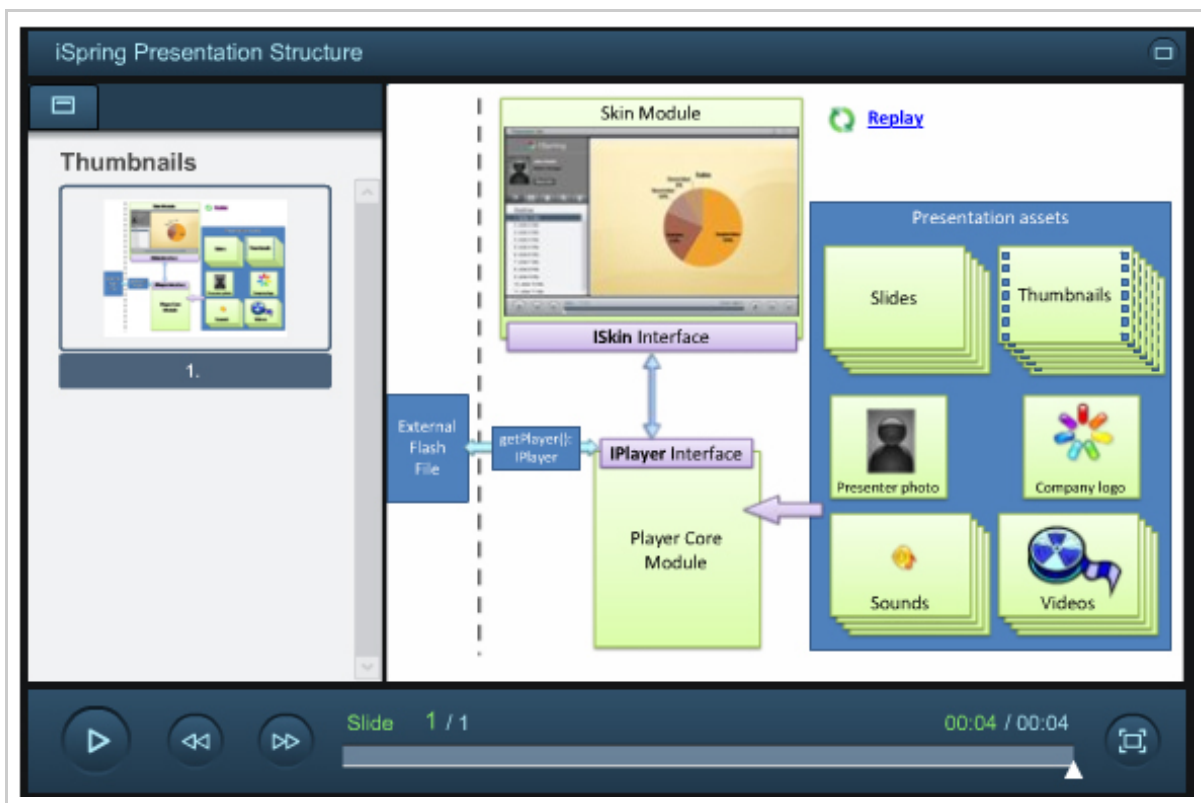
effects, transition effects and sounds. Player core is a blackbox providing external Flash files and skin modules with programmatic navigation and playback control via [ActionScript API](#).

In order to load and play presentation slides the core needs a target MovieClip object where slide assets are loaded. Since the location of this MovieClip depends on skin user interface, each skin module exposes the [ISkin](#) interface to inform the core about slide assets target movie clip.

### 1.5.1.3. Skin module

Skins are Flash modules which main responsibility is providing the player core with the MovieClip where slide assets should be loaded and displayed. Skins also provide viewers with navigation and playback facilities. Skins interacts with the player core via ActionScript API provided by player core.

There is a set of predefined skin modules with iSpring products. Some of them provide simple user interface while other skins provide full featured customizable User Interface.



It is possible to develop a customized skin module using Player Core ActionScript API functionality. You can find more information in the [Skin development article](#).

#### 1.5.1.4. References

[IPlayer Interface](#)

[ISkin Interface](#)

[Skin development](#)

[Flash Presentation ActionScript API Reference](#)

### 1.5.2. Flash Presentation ActionScript API Reference

#### 1.5.2.1. fsplayer.api.CPresentationLoader Class

This class extends the standard **MovieClipLoader** class available since Adobe Flash Player 7 and simplifies the process of iSpring generated Flash movies loading.

This class is not a part of the ActionScript API provided by generated Flash movies. A source code of this class is included into the **fsplayer.api** package.

The following methods and properties are provided by the **CPresentationLoader** class:

| Methods  |   |
|--|---|
| Method   | Description   |
| CPresentationLoader()  | Class constructor. It is used to create instances of the CPresentationLoader class  |
| setPlayerListener(listener: <a href="#">IPlayerListener</a> ):Void | This method passes the <a href="#">IPlayerListener</a> interface to the loader. The passed listener object will receive a notification when player is initialized |
| loadClip(url:String, target:MovieClip):Void                        | This method loads iSpring generated presentation into the specified movie clip  |

| Properties |                         |  |
|------------|-------------------------|--|
| Property   | Type                    | Description  |
| player     | <a href="#">IPlayer</a> | Returns the reference to the player in loaded Flash presentation |

|             |        |  |
|-------------|--------|--|
| bytesLoaded | Number | Returns number of loaded bytes of the Flash presentation |
| bytesTotal  | Number | Returns total number of bytes of the Flash presentation  |

## Samples

The following example illustrates the usage of the **CPresentationLoader** class.

```
import fsplayer.api.*;
var loader:CPresentationLoader = new CPresentationLoader();

// this object (in our case it is _root MovieClip) will listening to the
// player
loader.setPlayerListener(this);

// load generated Flash presentation
loader.loadClip("presentation.swf", presentation);

// this method is invoked when player gets initialized
function onPlayerInit(p:IPlayer):Void
{
    // add your extra initialization code here
}
```

## See also

[IPlayer Interface](#)

[IPlayerListener Interface](#)

[Flash Presentation ActionScript API Reference](#)

### 1.5.2.2. fsplayer.api.IAnimationStep Interface

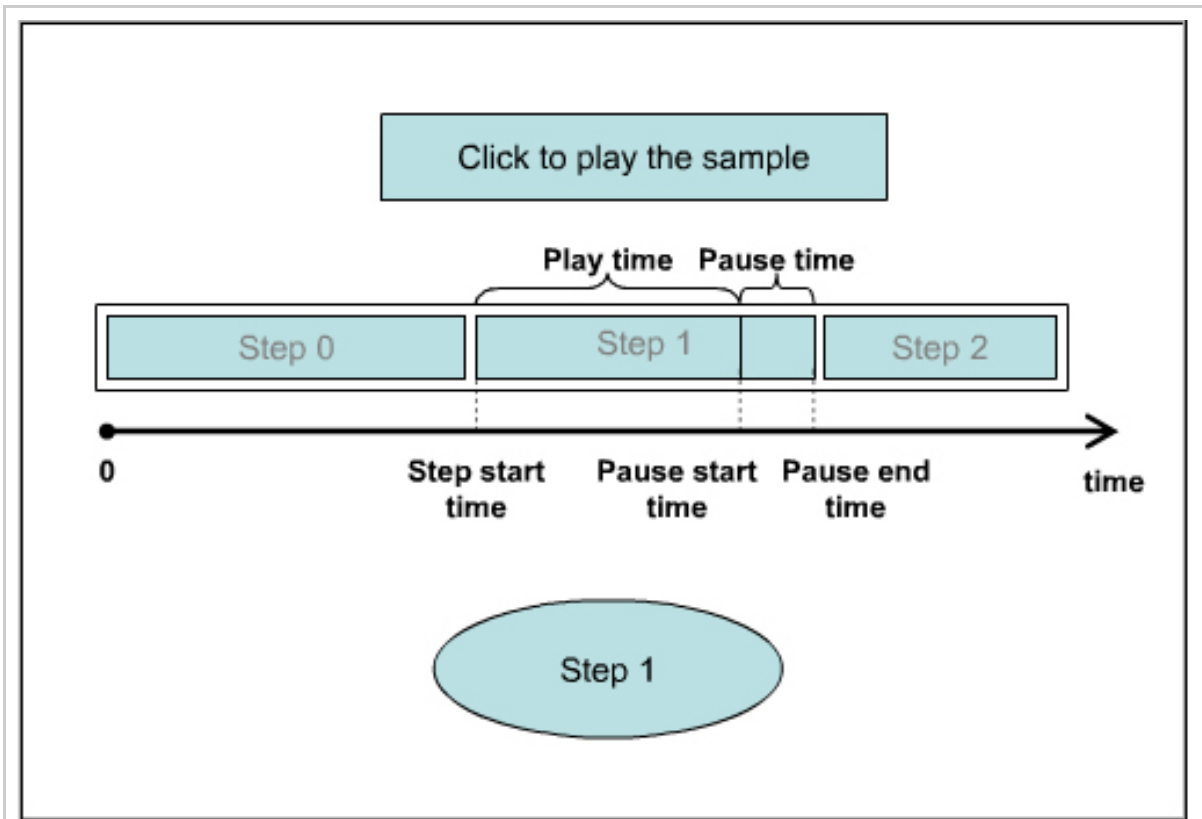
The **IAnimationStep** Interface provides the information about animation step timings. It exposes the following methods:

| Methods               |   |
|-----------------------|---|
| Method                | Description   |
| getPlayTime():Number  | Returns animation step play time (see the <a href="#">remarks</a> section)  |
| getPauseTime():Number | Returns animation step pause time (see the <a href="#">remarks</a> section) |

|                            |   |
|----------------------------|---|
| getStartTime():Number      | Returns animation step start time (see the <a href="#">remarks</a> section)       |
| getPauseStartTime():Number | Returns animation step pause start time (see the <a href="#">remarks</a> section) |
| getPauseEndTime():Number   | Returns animation step pause end time (see the <a href="#">remarks</a> section)   |

**Remarks**

Each slide consists of one or more animation steps with the following structure.



**Animation step structure**

An Animation step starts playing at the **Step start time**. Animations are playing for the **Play time**. Then, at the **Pause start time**, animations are suspended for the **Pause time** period till

the **Pause end time**. Pause time depends on slide timings that were set using "Record narration" or "Rehearse timings" menu of PowerPoint.

```
stepStartTime0 = 0
pauseStartTimei = stepStartTimei + playTimei
pauseEndTimei = pauseStartTimei + pauseTimei
stepStartTimei+1 = pauseEndTimei
```

If the slide has no timings (i.e. user has to click slide background to advance mouse-driven animations), **Pause time** and **Pause end time** are undefined. In this case returned timings are as follows:

```
pauseStartTimei = stepStartTimei + playTimei
pauseEndTimei = undefined
stepStartTimei+1 = pauseStartTimei
```

#### See also

[IAnimationSteps Interface](#)

[Flash Presentation ActionScript API Reference](#)

#### 1.5.2.3. fsplayer.api.IAnimationSteps Interface

The **IAnimationSteps** Interface provides the information about animation steps on the slide.

| Methods   |   |
|---|---|
| Method  | Description   |
| getStepsCount():Number                                | Returns number of animation steps on the slide  |
| getDuration():Number                                  | Returns the total duration of animation steps in seconds. The duration of animation steps is equal to the duration of the slide |
| getStep(index:Number): <a href="#">IAnimationStep</a> | Returns the <a href="#">IAnimationStep</a> Interface providing the information about animation step timings                     |

#### Remarks

Slide animation timeline consists of one or more animation steps. Animation step is a sequence of animation effects playing without mouse click requirement.

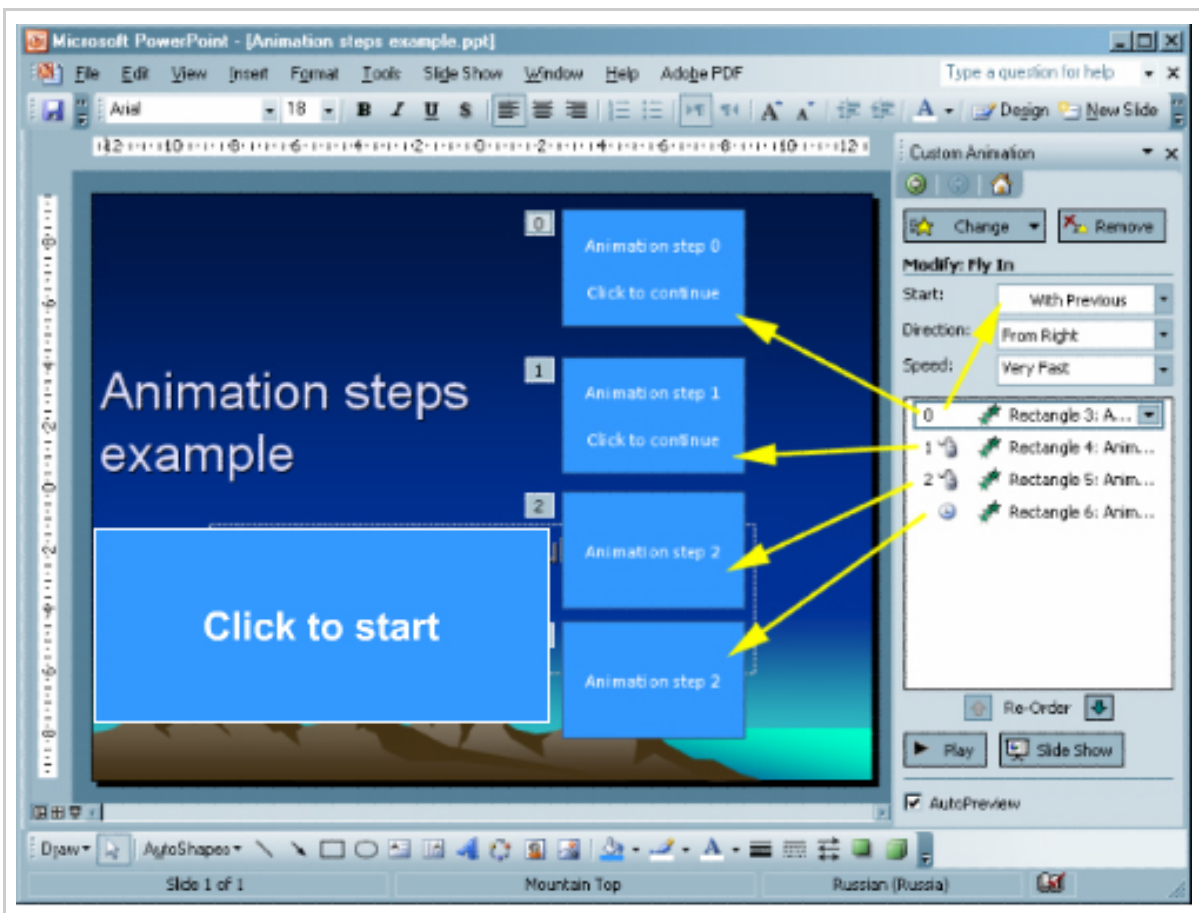
Slide background, static shapes, and shapes animated before the first mouse driven animation

are playing at the animation step 0. So there is at least 1 animation step in each slide.

Each mouse-driven animation along with the subsequent non mouse-driven animations are playing at the separate animation step. In other words, the number of animation steps is equal to number of on-click animations plus 1.

### Example

There are 4 animation effects in the following slide. They are represented by 3 animation steps in the generated Flash presentation (click to start animations).



- Effect 1 starts "With previous" animation effect. It is played at animation step 0
- Effects 2 and 3 start "On Click". They are played at animation steps 1 and 2
- Effect 4 starts "After previous" animation effect. It is played at animation step 2 after Effect 3.

**See also**[IAnimationStep Interface](#)[ISlideInfo Interface](#)[Flash Presentation ActionScript API Reference](#)**1.5.2.4. fsplayer.api.ICompanyInfo Interface**

The **ICompanyInfo** Interface provides the following methods for retrieving company information:

| Methods                                 |   |
|---|---|
| Method                                  | Description   |
| hasLogo():Boolean                       | Indicates whether the company has logo  |
| getLogo(): <a href="#">ICompanyLogo</a> | Returns the <a href="#">ICompanyLogo</a> interface providing information about company logotype |

**See also**[ICompanyLogo Interface](#)[IPresentationInfo Interface](#)[Flash Presentation ActionScript API Reference](#)**1.5.2.5. fsplayer.api.ICompanyLogo Interface**

The **ICompanyInfo** Interface provides the following methods for retrieving company logo information:

| Methods   |  |
|---|--|
| Method  | Description  |
| load(target:MovieClip, <i>[listener:<a href="#">ICompanyLogoLoadingListener</a>]</i> ):Void | Loads company logo into the specified MovieClip. The <b>listener</b> is an optional parameter which will receive notification when company logo is loaded  |
| getHyperlinkURL():String  | Returns a web page URL, for example, the company web site address, which should be opened when user clicks the company logo.<br><b>Note:</b> An empty string indicates that no hyperlink is attached to the company logo |

|                             |  |
|-----------------------------|--|
| getHyperlinkTarget():String | Returns the target window name ("_self", "_blank", "_parent", "_top", or custom window name) where company logo hyperlink should be opened |
|-----------------------------|--|

**See also**

[ICompanyInfo Interface](#)

[ICompanyLogoLoadingListener Interface](#)

[Flash Presentation ActionScript API Reference](#)

**1.5.2.6. fsplayer.api.ICompanyLogoLoadingListener Interface**

The **ICompanyLogoLoadingListener** Interface is used to receive the notification that company logo has been loaded. This interface exposes the following methods.

| Methods                      |   |
|------------------------------|---|
| Method                       | Description                                   |
| onCompanyLogoLoadInit():Void | Invoked when the company logo has been loaded |

This interface is mainly used for the development of a custom skin displaying company logo. It allows getting a notification that the company logo has been loaded in order to update skin layout at the proper time.

Usually you need to create a class implementing this interface and pass its instance to the **load()** method of [ICompanyLogo](#) interface.

**Sampes**

The following example illustrates how to load and place the company logo within target MovieClip.

```
import fsplayer.api.*;

class MyCompanyLogoPanel implements ICompanyLogoLoadingListener
{
    private var m_target:MovieClip;
    private var m_logo:MovieClip;

    MyCompanyLogoPanel(target:MovieClip, logo:ICompanyLogo)
    {
        m_target = target;
        m_logo = target.createEmptyMovieClip("logo",
target.getNextHighestDepth());
    }
}
```

```

        logo.load(target, this);
    }

    // implementation of
    ICompanyLogoLoadingListener.onCompanyLogoLoadInit() method
    function onCompanyLogoLoadInit():Void
    {
        // company logo has been loaded. We need to update its position
        with the target MovieClip
        centerLogo();
    }

    function centerLogo()
    {
        var logoWidth:Number = m_logo._width;
        var logoHeight:Number = m_logo._height;

        m_logo._x = (m_target._width - logoWidth) / 2;
        m_logo._y = (m_target._height - logoHeight) / 2;
    }
}

```

**See also**[ICompanyLogo Interface](#)[Flash Presentation ActionScript API Reference](#)**1.5.2.7. fsplayer.api.IPlaybackListener Interface**

There are various events occurring during presentation playback. The playback controller notifies external objects about these events via **IPlaybackListener** interface. This interface defines a set of methods which are invoked by playback controller when the corresponding events occur. These methods are as follows.

| Methods                                       |  |
|---|--|
| Method  | Description  |
| onPausePlayback():Void                        | Invoked when presentation playback is suspended              |
| onStartPlayback():Void                        | Invoked when presentation playback is resumed                |
| onAnimationStepChanged(stepIndex:Number):Void | Invoked when slide animation step is changed during playback |
| onSlidePositionChanged(position:Number):Void  | Invoked when playback position is changed during playback    |

|  |   |
|--|---|
| onCurrentSlideIndexChanged(slideIndex:Number):Void     | Invoked when slide switching occurs   |
| onSlideLoadingComplete(slideIndex:Number):Void         | Invoked when slide loading is completed   |
| onPresentationPlaybackComplete():Void                  | Invoked at the end of presentation playback   |
| onKeyboardFocusStateChanged(acquireFocus:Boolean):Void | Invoked when interactive element of the presentation (e.g. a text field of a Quiz) acquires or loses keyboard focus |

In order to receive event notifications you need to create a class which implements **IPlaybackListener** interface and pass the instance of this class to the **addListener** method of the [IPresentationPlaybackController](#) interface.

### Samples

The following example illustrates the simplest implementation of **IPlaybackListener** interface.

Create **MyPlaybackListener.as** source file with the following content.

```
import fsplayer.api.*;

class MyPlaybackListener implements IPlaybackListener
{
function onPausePlayback():Void
{
trace("Presentation has been paused");
}

function onStartPlayback():Void
{
trace("Presentation playback was started");
}

function onCurrentSlideIndexChanged(slideIndex:Number):Void
{
trace("Current slide index has been changed to " + slideIndex);
}

function onSlideLoadingComplete(slideIndex:Number):Void
{
trace("Slide " + slideIndex + " has been just loaded");
}

function onSlidePositionChanged(position:Number):Void
{

```

```
trace("Current slide playback position was changed to " + position);
}

function onPresentationPlaybackComplete():Void
{
trace("Presentation playback has been completed");
}

function onAnimationStepChanged(stepIndex:Number):Void
{
trace("Animation step index has been changed to " + stepIndex);
}
}
```

#### Usage of **MyPlaybackListener** class:

```
import fsplayer.api.*;
import MyPlaybackListener;

var presentation:MovieClip =
this.createEmptyMovieClip("presentation", 1);

var loader:CPresentationLoader = new CPresentationLoader()
loader.setPlayerListener(this);
loader.loadClip("c:\\test.swf", presentation);

function onPlayerInit(player:IPlayer)
{
var listener:MyPlaybackListener = new MyPlaybackListener();
player.getPlaybackController().addListener(listener);
}
```

Instead of implementing **IPlaybackListener** interface it is also possible to create an Object instance with a subset of IPlaybackListener functions and pass it to the **addListener()** method as it is shown in the following example:

```
import fsplayer.api.*;

var presentation:MovieClip =
this.createEmptyMovieClip("presentation", 1);

var loader:CPresentationLoader = new CPresentationLoader()
loader.setPlayerListener(this);
loader.loadClip("c:\\test.swf", presentation);

function onPlayerInit(player:IPlayer)
{
var listener = new Object();
listener.onCurrentSlideIndexChanged = function(index:Number):Void
{
trace("Current slide index has been changed to " + index);
}
```

```

    }
    player.getPlaybackController().addListener(listener);
}

```

**See also**

[IPresentationPlaybackController Interface](#)  
[Flash Presentation ActionScript API Reference](#)

**1.5.2.8. fsplayer.api.IPlayer Interface**

iSpring generated Flash presentations contain an embedded player which plays slide animations, transition effects and sounds. This player exposes an ActionScript 2.0 API available to external Flash applications.

The **IPlayer** interface is provided by Flash presentations generated with iSpring. It exposes the following methods:

| Methods   |   |
|---|---|
| Method  | Description   |
| isInitialized():Boolean   | Returns a Boolean value indicating whether the player was initialized.<br>The following methods return an <b>undefined</b> value until the player gets initialized: <ul style="list-style-type: none"> <li>• <a href="#">getPresentationInfo()</a></li> <li>• <a href="#">getPlaybackController()</a></li> <li>• <a href="#">getSoundController()</a></li> <li>• <a href="#">getSettings()</a></li> </ul> |
| addListener(listener: <a href="#">IPlayerListener</a> ):Void          | Adds an external listener object implementing the <a href="#">IPlayerListener</a> interface to the player. The passed listener object will receive notifications from the player  |
| removeListener(listener: <a href="#">IPlayerListener</a> ):Void       | Removes the listener object added earlier with <b>addListener()</b> method  |
| getPlaybackController(): <a href="#">IPresentationPlaybackControl</a> | Returns the <a href="#">IPresentationPlaybackController</a> interface providing presentation playback control and navigation facilities   |

|  |   |
|--|---|
| getSoundController(): <a href="#">ISoundController</a>   | Returns the <a href="#">ISoundController</a> interface providing sound volume control facilities  |
| getPresentationInfo(): <a href="#">IPresentationInfo</a> | Returns the <a href="#">IPresentationInfo</a> interface providing the presentation content information  |
| getSettings():Object                                     | Returns an Object storing playback and navigation settings as well as skin related settings (see the <a href="#">remarks</a> section)<br>This method is mainly used for the custom skins development. In most cases you don't need to use it. |
| getSkin(): <a href="#">ISkin</a>                         | Returns the <a href="#">ISkin</a> interface of the player   |

#### Remarks

An object returned by the **getSettings()** method contains a set of implementation specific properties. Currently, this object has the following structure:

- **navigation** (Object) - the set of navigation properties:
  - **keyboard** (Object) - keyboard navigation properties:
    - **enabled** (String) - determines whether keyboard navigation is enabled ("1") or not
  - **mouse** (Object) - mouse navigation properties
    - **enabled** (String) - determines whether navigation by mouse clicks on the slide background is enabled ("1") or not
    - **target** (String) - determines whether the mouse click navigation is performed by slides ("slide") or steps ("step")
- **colorScheme** (Object) - color scheme settings:
  - **primaryColor** (String) - this property is obsolete and must not be used in new skins
- **appearance** (Object) - appearance settings:
  - **fullScreen** (String) - determines whether player window is opened full screen ("1") or not
- **playback** (Object) - playback settings:
  - **autoStart** (String) - determines whether the player starts presentation playback automatically ("1" or undefined) or not
- **webLinks** (Object) - the set of external web links related to the presentation:
  - **slideAreaLink** (Object) - web link related to the slide area
    - **href** (String) - web page reference, e.g. "<http://www.ispringsolutions.com>"

- **target** (String) - target window, e.g. "*\_self*", "*\_blank*", "*\_parent*" or custom window name
- **presentationEndLink** (Object) - web link related to the presentation finish event
  - **href** (String) - web page reference, e.g. "*http://www.ispringsolutions.com*"
  - **target** (String) - target window, e.g. "*\_self*", "*\_blank*", "*\_parent*" or custom window name

**See also**

- [IPlayerListener Interface](#)
- [IPresentationPlaybackController Interface](#)
- [ISoundController Interface](#)
- [IPresentationInfo Interface](#)
- [ISkin Interface](#)
- [Flash Presentation ActionScript API Reference](#)

**1.5.2.9. fsplayer.api.IPlayerCore Interface**

The **IPlayerCore** interface provides advanced facilities for customizing iSpring player functionality. **This interface should be used by skilled developers only.** An improper use of this interface may lead to the player malfunction.

| Methods   |   |
|---|---|
| Method  | Description   |
| setExternalPresenterVideoPlayer(player: <b>IVideoPlayer</b> ) | Method allows replacing of the standard presenter video player with the custom one during player initialization. <b>Note:</b> this method should be called before player gets initialized, in the <b>onPlayerPreInit</b> event of the <a href="#">IPlayerListenerEx</a> . |

**See also**

- [IPlayerListenerEx Interface](#)
- [IPlayerEx Interface](#)

**1.5.2.10. fsplayer.api.IPlayerEx Interface**

Extends the [IPlayer](#) interface providing access to the advanced functionality of the presentation player.

The **IPlayerEx** interface has following method:

| Methods                                      |  |
|--|--|
| Method                                       | Description  |
| getPlayerCore(): <a href="#">IPlayerCore</a> | Returns the <a href="#">IPlayerCore</a> interface providing access to the player core functionality. |

**See also**

[IPlayer Interface](#)

[IPlayerCore Interface](#)

[Flash Presentation ActionScript API Reference](#)

#### 1.5.2.11. fsplayer.api.IPlayerListener Interface

The **IPlayerListener** interface is used to receive the notification that the player has been initialized. Until the player gets initialized it is not possible to control presentation playback or get presentation information.

The **IPlayerListener** interface declares the following method.

| Methods   |  |
|---|--|
| Method  | Description  |
| onPlayerInit(player: <a href="#">IPlayer</a> ):Void | This method is invoked when the player initialization is complete. |

Usually a class implementing the **IPlayerListener** interface is created.

```
import fsplayer.api.IPlayerListener;
import fsplayer.api.IPlayer;

class MyPlayerListener implements IPlayerListener
{
    // ...

    function onPlayerInit(player:IPlayer):Void
    {
        // write your player initialization handler here
    }

    // ...
}
```

Then the instance of `MyPlayerListener` object is passed to `addListener()` method of the [IPlayer](#) interface or to the `setPlayerListener()` method of the [CPresentationLoader](#) class which simplifies this process.

```
import fsplayer.api.IPlayer;
import MyPlayerListener;
import fsplayer.api.CPresentationLoader;

// create instances of the presentation loader and the player listener
objects
var loader:CPresentationLoader = new CPresentationLoader();
var listener:MyPlayerListener = new MyPlayerListener();

// pass the listener object to the loader
loader.setPlayerListener(listener);

// create a MovieClip instance where the presentation will be loaded
var presentationMC:MovieClip = this.createEmptyMovieClip("presentation",
1);

// load the presentation
loader.loadClip("presentation.swf", presentationMC);
```

**See also**

- [IPlayer Interface](#)
- [Flash Presentation ActionScript API Reference](#)

### 1.5.2.12. fsplayer.api.IPlayerListenerEx Interface

The **IPlayerListenerEx** interface is used to receive advanced notification that the player has been preinitialized.

The **IPlayerListenerEx** interface declares the following method.

| Methods   |   |
|---|---|
| Method  | Description   |
| <code>onPlayerPreInit(player:<a href="#">IPlayerEx</a>):Void</code> | This method is invoked before the player initialization is completed. |

**See also**

- [IPlayerEx Interface](#)
- [IPlayerCore Interface](#)
- [Flash Presentation ActionScript API Reference](#)

### 1.5.2.13. fsplayer.api.IPresentationInfo Interface

The **IPresentationInfo** Interface provides the following methods for retrieving presentation information:

| Methods  |  |
|--|--|
| Method   | Description  |
| getTitle():String                                      | Returns presentation title   |
| getSlides(): <a href="#">ISlidesCollection</a>         | Returns <a href="#">ISlidesCollection</a> interface providing information about presentation slides                  |
| getSlideWidth():Number                                 | Returns slide width in pixels  |
| getSlideHeight():Number                                | Returns slide height in pixels   |
| hasThumbnails():Boolean                                | Indicates whether the presentation contains thumbnail images   |
| getThumbnails(): <a href="#">IThumbnailsCollection</a> | Returns <a href="#">IThumbnailsCollection</a> interface that allows to get information about presentation thumbnails |
| hasPresenter():Boolean                                 | Indicates whether the presentation has got presenter information   |
| getPresenterInfo(): <a href="#">IPresenterInfo</a>     | Returns the <a href="#">IPresenterInfo</a> interface which provides the information about presentation presenter     |
| getFrameRate():Number                                  | Returns presentation frame rate (number of frames per second)  |
| getDuration():Number                                   | Returns presentation duration in seconds   |
| hasCompanyInfo():Boolean                               | Indicates whether the presentation contains company information  |
| getCompanyInfo(): <a href="#">ICompanyInfo</a>         | Returns the <a href="#">ICompanyInfo</a> interface providing company information                                     |

|  |   |
|--|---|
| hasReferences():Boolean                                | Indicates whether the presentation contains references  |
| getReferences(): <a href="#">IReferencesCollection</a> | Returns the <a href="#">IReferencesCollection</a> interface providing the information about presentation references |
| getVisibleDuration():Number                            | Returns duration of visible slides in seconds   |
| getPresenters(): <a href="#">IPresentersCollection</a> | Returns the <a href="#">IPresentersCollection</a> interface providing the information about presentation presenters |

**See also**[ICompanyInfo Interface](#)[IPresentersCollection](#)[IPresenterInfo Interface](#)[IReferencesCollection Interface](#)[ISlidesCollection Interface](#)[IThumbnailsCollection Interface](#)[Flash Presentation ActionScript API Reference](#)**1.5.2.14. fsplayer.api.IPresentationPlaybackController Interface**

The **IPresentationPlaybackController** interface provides methods for presentation playback and navigation control. This interface can be obtained by calling the **getPlaybackController()** method on [IPlayer](#) interface.

| Methods   |  |
|---|--|
| Method  | Description  |
| addListener(listener: <a href="#">IPlaybackListener</a> ):Void    | Adds an external listener object implementing the <a href="#">IPlaybackListener</a> interface; the passed object will be notified about presentation playback events |
| removeListener(listener: <a href="#">IPlaybackListener</a> ):Void | Removes previously added listener object. The passed object won't receive notifications from the player any more   |
| isPlaying():Boolean   | Returns a Boolean value indicating whether the presentation is playing or not  |

|  |  |
|--|--|
| <code>getCurrentSlideDuration():Number</code>                              | Returns current slide duration in seconds  |
| <code>play():Void</code>   | Resumes presentation playback  |
| <code>pause():Void</code>  | Suspends presentation playback   |
| <code>gotoNextSlide([autoStart:Boolean = true]):Void</code>                | Switches to the next presentation slide and starts its playback.<br><b>Note:</b> if the <code>autoStart</code> parameter was set to false, slide playback would not be started after switching                                 |
| <code>gotoPreviousSlide([autoStart:Boolean = true]):Void</code>            | Switches to the previous presentation slide and starts its playback.<br><b>Note:</b> if the <code>autoStart</code> parameter was set to false, slide playback would not be started after switching                             |
| <code>gotoLastViewedSlide([autoStart:Boolean = true]):Void</code>          | Switches to the previously viewed slide and starts its playback.<br><b>Note:</b> if the <b>autoStart</b> parameter was set to false, slide playback would not be started after switching                                       |
| <code>getCurrentSlideIndex():Number</code>                                 | Returns a zero-based index of currently playing slide  |
| <code>gotoSlide(slideIndex:Number, [autoStart:Boolean = true]):Void</code> | Switches to the given slide of the presentation and starts its playback<br><b>Notes:</b> Slide index is zero-based.<br>If the <b>autoStart</b> parameter was set to false, slide playback would not be started after switching |
| <code>getCurrentSlidePlaybackPosition():Number</code>                      | Returns playback position within current slide.<br><b>Note:</b> playback position is a number between 0 and 1  |
| <code>pauseCurrentSlideAt(position:Number):Void</code>                     | Suspends currently playing slide at the given position<br><b>Note:</b> position is a number in between 0 and 1   |
| <code>playCurrentSlideFrom(position:Number):Void</code>                    | Restarts current slide from the given position.<br><b>Note:</b> position is a number between 0 and 1   |
| <code>seek(position:Number):Void</code>                                    | Seeks the current slide to the given position. After this operation slide playback gets suspended until the <b>endSeek()</b> method or other playback control method   |

|   |  |
|---|--|
|   | is called<br><b>Note:</b> position is a number between 0 and 1   |
| <code>endSeek([resumePlayback:Boolean = undefined]):Void</code>     | Exits from seeking mode and restores playback.<br><b>Note:</b> the <b>resumePlayback</b> parameter allows to start or pause slide after seeking explicitly   |
| <code>getCurrentStepIndex():Number</code>                           | Returns current animation step index   |
| <code>gotoNextStep():Void</code>                                    | Plays current slide from the beginning of the next animation step.<br><b>Note:</b> being called at the end of the slide this method plays presentation from the beginning of the next slide  |
| <code>gotoPreviousStep():Void</code>                                | Moves slide playback one step back. This method suspends slide playback at the end of the previous animation step.<br><b>Note:</b> being called on the start slide step this method causes playback switching to the end of the previous slide |
| <code>setAnimationStepPause(pause:Number):Void</code>               | This method allows setting a pause between animation steps. If an <b>undefined</b> value is passed as pause value, animation steps will not be played automatically  |
| <code>playFromStep(stepIndex:Number):Void</code>                    | Plays current slide from the beginning of the specified animation step   |
| <code>pauseAtStepStart(stepIndex:Number):Void</code>                | Suspends current slide at the beginning of the specified animation step  |
| <code>pauseAtStepEnd(stepIndex:Number):Void</code>                  | Suspends current slide at the end of specified animations step   |
| <code>enableAutomaticSlideSwitching(enableSwitching:Boolean)</code> | Enables or disables automatic switching to the next slide.<br><b>Note:</b> this method affects automatically switching slides only.  |
| <code>getAutomaticSlideSwitching():Boolean</code>                   | Returns true if automatic slide switching has been enabled, false otherwise.   |

|   |  |
|---|--|
|   | <b>Note:</b> automatic slide switching is enabled by default   |
| getCurrentVisibleSlideIndex():Number  | Returns index of currently playing slide within visible slides.<br>Returns an undefined value if current slide is hidden.<br><b>Note:</b> Slide index is zero-based.   |
| gotoVisibleSlide(visibleSlideIndex:Number, [autoStart:Boolean = true]):Void | Switches to the given index within visible slides and starts its playback<br><b>Notes:</b> Slide index is zero-based.<br>If the <b>autoStart</b> parameter was set to false, slide playback would not be started after switching |
| gotoFirstSlide([autoStart:Boolean = true]):Void                             | Switches to the first presentation slide and starts its playback.<br><b>Note:</b> if the <b>autoStart</b> parameter was set to false, slide playback would not be started after switching  |
| gotoLastSlide([autoStart:Boolean = true]):Void                              | Switches to the last presentation slide and starts its playback.<br><b>Note:</b> if the <b>autoStart</b> parameter was set to false, slide playback would not be started after switching   |
| getNextSlideIndex():Number  | Returns next slide index or an undifined value in case if current slide is the latest one  |
| getPreviousSlideIndex():Number  | Returns previous slide index or an undifined value in case if current slide is the first one   |
| isNamedSlideShowPlaying():Boolean   | Returns Boolean value that indicates whether some named slide show is playing  |

**See also**[IPlaybackListener Interface](#)[IPlayer Interface](#)[Flash Presentation ActionScript API Reference](#)**1.5.2.15. fsplayer.api.IPresentationPlaybackControllerEx Interface**

Extends the [IPresentationPlaybackController](#) interface providing information about currently playing quiz.

The **IPresentationPlaybackControllerEx** interface has following method:

| Methods                          |   |
|----------------------------------|---|
| Method                           | Description   |
| getActiveQuiz(): <b>IQuizApi</b> | Returns the IQuizApi interface that allows to get information about currently shown quiz.<br><b>Note:</b> the IQuizApi interface is used for internal purposes only and is not available for public access in current version of the API. |

**See also**

[IPresentationPlaybackController](#)

[IPlaybackListener Interface](#)

[Flash Presentation ActionScript API Reference](#)

#### 1.5.2.16. fsplayer.api.IPresenterInfo Interface

The **IPresenterInfo** interface provides the following methods for presenter information retrieving:

| Methods                                     |  |
|---|--|
| Method                                      | Description  |
| getName():String                            | Returns presenter name   |
| getTitle():String                           | Returns presenter title  |
| getBiographyText():String                   | Returns brief information about the presenter  |
| getEmail():String                           | Returns presenter email address  |
| getWebSite():String                         | Returns presenter web site address   |
| hasPhoto():Boolean                          | Indicates whether the presenter has photo  |
| getPhoto(): <a href="#">IPresenterPhoto</a> | Returns the <a href="#">IPresenterPhoto</a> interface which allows to load presenter photo |
|   |  |

|  |   |
|--|---|
| getIndex():Number                          | Returns index within presenters collection  |
| hasCompany():Boolean                       | Indicates whether the presenter has company   |
| getCompany(): <a href="#">ICompanyInfo</a> | Returns the <a href="#">ICompanyInfo</a> interface providing information about presenter custom company |

**See also**[ICompanyInfo Interface](#)[IPresentationInfo Interface](#)[IPresenterPhoto Interface](#)[Flash Presentation ActionScript API Reference](#)**1.5.2.17. fsplayer.api.IPresenterPhoto Interface**

The **IPresenterPhoto** interface allows to load presenter photo.

| Methods  |   |
|--|---|
| Method   | Description   |
| load(target:MovieClip, listener: <a href="#">IPresenterPhotoLoadingListener</a> ):Void | This method loads presenter photo into the specified target MovieClip (see the <a href="#">remarks</a> section for more details). The <b>listener</b> will be notified when presenter photo is loaded |

**Remarks**

The **IPresenterPhoto** interface provides a unified way of the presenter photo loading no matter if the presentation has been exported into a solid Flash file, or into a set of Flash files with a player.

It is highly recommended that the **target** MovieClip resides within the skin MovieClip.

**See also**[IPresenterInfo Interface](#)[IPresenterPhotoLoadingListener Interface](#)[Flash Presentation ActionScript API Reference](#)**1.5.2.18. fsplayer.api.IPresenterPhotoLoadingListener Interface**

The **IPresenterPhotoLoadingListener** Interface is used to receive the notification that the presenter photo has been loaded. This interface exposes the following methods.

| Methods                         |   |
|---------------------------------|---|
| Method                          | Description   |
| onPresenterPhotoLoadInit():Void | This method is invoked when the presenter photo has been loaded.<br>The <b>load()</b> method of the <a href="#">IPresenterPhoto</a> interface loads the presenter photo into the specified target MovieClip |

**See also**

[IPresenterPhoto Interface](#)  
[Flash Presentation ActionScript API Reference](#)

#### 1.5.2.19. fsplayer.api.IPresentersCollection Interface

The **IPresentersCollection** Interface represents a collection of the presentaiton presenters. It exposes the following methods:

| Methods  |   |
|--|---|
| Method   | Description   |
| getCount():Number  | Returns number of presenters in the collection  |
| getPresenter(index:Number): <a href="#">IPresenterInfo</a> | Returns the <a href="#">IPresenterInfo</a> interface providing the information about the specified presenter.<br><b>Note:</b> the <b>index</b> parameter is an integer value from 0 to (number of presenters - 1) |

**See also**

[IPresenterInfo Interface](#)  
[IPresentationInfo Interface](#)  
[Flash Presentation ActionScript API Reference](#)

#### 1.5.2.20. fsplayer.api.IPresenterVideo Interface

The **IPresenterVideo** interface provides the following methods for presenter video information retrieving:

| Methods            |  |
|--------------------|--|
| Method             | Description                            |
| getWidth():Number  | Returns width of the presenter video.  |
| getHeight():Number | Returns height of the presenter video. |

**See also**[IPresenterInfo Interface](#)[IPresenterVideoListener Interface](#)[Flash Presentation ActionScript API Reference](#)**1.5.2.21. fsplayer.api.IPresenterVideoController Interface**

The **IPresenterVideoController** Interface provides the following methods for presenter video control.

| Methods   |   |
|---|---|
| Method  | Description   |
| addListener(listener: <a href="#">IPresenterVideoListener</a> ):Void    | Adds an external object implementing <a href="#">IPresenterVideoListener</a> interface which will receive presenter video controller notifications. |
| removeListener(listener: <a href="#">IPresenterVideoListener</a> ):Void | Removes the previously added listener. The passed object will stop receiving presenter video controller notifications.                              |
| getCurrentVideo(): <a href="#">IPresenterVideo</a>                      | Returns current presenter video   |

**See also**[IPresenterVideo Interface](#)[IPresenterVideoListener Interface](#)[Flash Presentation ActionScript API Reference](#)**1.5.2.22. fsplayer.api.IPresenterVideoListener Interface**

The **IPresenterVideoListener** interface defines a method invoked by presenter video controller when a particular events occur during presenter video playback.

| Methods  |  |
|--|--|
| Method   | Description  |
| onChangePresenterVideo(video: <a href="#">IPresenterVideo</a> ):Void | This method is invoked when presenter video is changed. Use the <b>getCurrentVideo()</b> method of the passed <a href="#">IPresenterVideoController</a> interface to get new video |

In order to receive presenter video notifications you need to create a class which implements [IPresenterVideoListener](#) interface and pass it to the **addListener()** method of the [IPresenterVideoController](#) Interface.

**See also**

- [IPresenterVideo](#)
- [IPresenterVideoController Interface](#)
- [Flash Presentation ActionScript API Reference](#)

**1.5.2.23. fsplayer.api.IReferenceInfo Interface**

The **IReferenceInfo** Interface provides the following methods for retrieving reference information:

| Methods            |   |
|--------------------|---|
| Method             | Description   |
| getTitle():String  | Returns reference title   |
| getURL():String    | Returns reference URL   |
| getTarget():String | Returns target window name where the reference should be opened |

**See also**

- [IReferencesCollection Interface](#)
- [Flash Presentation ActionScript API Reference](#)

**1.5.2.24. fsplayer.api.IReferencesCollection Interface**

The **IReferencesCollection** Interface provides the following methods for retrieving the

presentation references information:

| Methods  |   |
|--|---|
| Method   | Description   |
| getCount():Number  | Number of references in the collection  |
| getReference(index:Number): <a href="#">IReferenceInfo</a> | Returns <a href="#">IReferenceInfo</a> interface providing the information about the specified reference.<br><b>Note:</b> the <b>index</b> parameter is an integer value from 0 to (number of references - 1) |

**See also**

[IReferenceInfo Interface](#)

[IPresentationInfo Interface](#)

[Flash Presentation ActionScript API Reference](#)

#### 1.5.2.25. fsplayer.api.ISlideInfo Interface

The **ISlideInfo** Interface provides information about the particular slide of the presentation such as title, duration, text, etc. It exposes the following methods.

| Methods  |   |
|--|---|
| Method   | Description   |
| isLoading():Boolean                                  | Returns Boolean value indicating whether the slide has been loaded completely                           |
| getTitle():String                                    | Returns slide title   |
| getAnimationSteps(): <a href="#">IAnimationSteps</a> | Returns <a href="#">IAnimationSteps</a> interface providing the information about slide animation steps |
| getDuration():Number                                 | Returns slide duration in seconds   |
| getNotesText():String                                | Returns text of the slide notes   |
| getStartTime():Number                                | Returns slide start time within the presentation timeline   |

|                                 |   |
|---------------------------------|---|
| getEndTime():Number             | Returns slide end time within the presentation timeline   |
| getStartStepIndex():Number      | Returns slide start step index within the presentation  |
| getEndStepIndex():Number        | Returns slide end step index within the presentation  |
| getSlideText():String           | Returns slide text in the <b>normalized form</b> (see the <a href="#">remarks</a> section)  |
| getNotesTextNormalized():String | Returns text of the slide notes in the <b>normalized form</b> (see the <a href="#">remarks</a> section)   |
| getTitleNormalized():String     | Returns slide title in the <b>normalized form</b> (see the <a href="#">remarks</a> section)   |
| getLevel():Number               | Returns slide level (0 - upper level)   |
| isHidden():Boolean              | Returns Boolean value that indicates whether the slide is hidden  |
| function getIndex():Number      | Returns slide index within presentation   |
| getVisibleIndex():Number        | Returns slide index within visible slides<br><b>Note:</b> This method returns an undefined value if the slide is hidden   |
| getVisibleStartTime:Number      | Returns slide start time within the presentation timeline for visible slides only<br><b>Note:</b> This method returns an undefined value if the slide is hidden |
| getVisibleEndTime:Number        | Returns slide end time within the presentation timeline for visible slides only<br><b>Note:</b> This method returns an undefined value if the slide is hidden   |
| getVisibleStartStepIndex:Number | Returns slide start step index within visible slides only<br><b>Note:</b> This method returns an undefined value if the slide is hidden                         |

|   |  |
|---|--|
| getVisibleEndStepIndex:Number                   | Returns slide end step index within visible slides only<br><b>Note:</b> This method returns an undefined value if the slide is hidden                    |
| getPresenterIndex():Number                      | Returns index of presenter associated with the slide<br><b>Note:</b> This method returns an undefined value if no presenter is associated with the slide |
| getResources(): <a href="#">ISlideResources</a> | Returns the <a href="#">ISlideResources</a> interface providing the information about resources contained in the slide                                   |
| getNotesHtml():String                           | Returns HTML representation of the slide notes   |

**Remarks**

**Normalized strings** are strings having all characters in the lower case with no extra spaces between words. This can simplify and speed up full text search implementation.

**See also**

[IAnimationSteps Interface](#)

[ISlidesCollection Interface](#)

[ISlideResources Interface](#)

[Flash Presentation ActionScript API Reference](#)

**1.5.2.26. fsplayer.api.ISlidesCollection Interface**

The **ISlidesCollection** interface represents a collection of presentation slides. It provides the following methods:

| Methods   |  |
|---|--|
| Method  | Description  |
| getSlidesCount():Number                                     | Returns number of slides in the presentation   |
| getSlideInfo(slideIndex:Number): <a href="#">ISlideInfo</a> | Returns <a href="#">ISlideInfo</a> Interface which represents an individual presentation slide with the specified slide index.<br><b>Note:</b> Slide index is zero-based |
|   |  |

|  |   |
|--|---|
| getVisibleSlidesCount():Number                       | Returns number of visible slides in the presentation  |
| getVisibleSlide(visibleSlideIndex:Number):ISlideInfo | Returns <a href="#">ISlideInfo</a> Interface which represents an individual visible slide with the specified slide index.<br><b>Note:</b> Slide index is zero-based |

**Remarks:**

There are 2 types of slides in the presentation:

- **Visible slides.** These are slides which are normally displayed during slide show
- **Hidden slides.** These are slides which were marked as hidden in PowerPoint during presentation creation. They are shown only by hyperlink from visible presentation slide.

**See also**

[IPresentationInfo Interface](#)

[ISlideInfo Interface](#)

[Flash Presentation ActionScript API Reference](#)

**1.5.2.27. fsplayer.api.ISlideThumbnail Interface**

The **ISlideThumbnail** interface provides the information about the particular presentation thumbnail and provides method for thumbnail loading into the specified MovieClip.

| Methods  |  |
|--|--|
| Method   | Description  |
| getSlideIndex():Number   | Returns slide index the thumbnail corresponds to   |
| load(target:MovieClip, [listener: <a href="#">ISlideThumbnailLoadingListener</a> ]):Void | Loads the thumbnail image into the target MovieClip. When the thumbnail is loaded, the notification will be sent to the passed <b>listener</b> interface. The <b>target</b> parameter is a MovieClip within the skin MovieClip (see the <a href="#">remarks</a> section) |

**Remarks**

When the presentation is wrapped into a skin which supports thumbnails (Classic, Crystal, Advanced or Lite), iSpring generates thumbnails in one of the following ways:

- When the presentation is exported into a solid Flash movie, iSpring stores Flash

presentation thumbnails as internal Flash movie assets. Such assets are created using **MovieClip.attachMovie()** method.

The reason for the **target** parameter of the **load()** method restriction is as follows: thumbnails, the skin and the player core are located within the same Flash file, so thumbnails assets are accessible within that Flash file only.

- When the presentation is exported into a set of Flash files with player, each thumbnail is saved into a separate Flash file. Such assets are loaded using **MovieClip.loadMovie()** method or via **MovieClipLoader** class.

The **ISlideThumbnail** interface provides skins with the unified way of internal and external thumbnails loading. Thumbnails are loaded in the proper way by player core. The core also notifies skin when the particular thumbnail is loaded via the [ISlideThumbnailLoadingListener](#) Interface.

#### See also

[ISlideThumbnailLoadingListener Interface](#)

[IThumbnailsCollection Interface](#)

[Flash Presentation ActionScript API Reference](#)

#### 1.5.2.28. fsplayer.api.ISlideThumbnailLoadingListener Interface

The **IPresenterThumbnailLoadingListener** Interface is used to receive the notification that the slide thumbnail has been loaded. This interface exposes the following methods:

| Methods   |   |
|---|---|
| Method  | Description   |
| onSlideThumbnailLoadInit(thumbnail: <a href="#">ISlideThumbnail</a> ) | Invoked when the thumbnail has been loaded. The <b>thumbnail</b> parameter can be used to determine the index of the loaded slide |

#### See also

[ISlideThumbnail Interface](#)

[Flash Presentation ActionScript API Reference](#)

#### 1.5.2.29. fsplayer.api.ISlideResources Interface

The **ISlideResources** interface provides the information about resources related to this presentation slide such as presenter video. The Interface exposes the following methods:

| Methods |
|---------|
|         |

| Method   | Description   |
|--|---|
| hasPresenterVideo():Boolean                          | Indicate whether the slide has a presenter video.   |
| getPresenterVideo(): <a href="#">IPresenterVideo</a> | Returns the <a href="#">IPresenterVideo</a> interface providing information about presenter video.<br>If there is no presenter video, an undefined value is returned. |

**See also**[IPresenterVideo Interface](#)**1.5.2.30. fsplayer.api.ISoundController Interface**

The **ISoundController** Interface provides the following methods for sound control.

| Methods  |  |
|--|--|
| Method   | Description  |
| addListener(listener: <a href="#">ISoundListener</a> ):Void    | Adds an external object implementing <a href="#">ISoundListener</a> interface which will receive sound controller notifications                                |
| removeListener(listener: <a href="#">ISoundListener</a> ):Void | Removes the previously added listener. The passed object will stop receiving sound controller notifications  |
| getVolume():Number   | Returns current sound volume.<br><b>Note:</b> sound volume is a number between 0 and 1   |
| setVolume(volume:Number):Void                                  | Sets sound volume.<br><b>Note:</b> sound volume is a number between 0 and 1  |
| tweak(parameterName:String, parameterValue:Object):Void        | Tweaks the implementation specific parameter of the sound controller.<br>See the list of currently supported parameters in the <a href="#">Remarks</a> section |
| getTweakableParameter(parameterName:String):Object             | Returns the value of the implementation specific parameter.<br>See the list of currently supported parameters in the   |

|  |                                 |
|--|---------------------------------|
|  | <a href="#">Remarks</a> section |
|--|---------------------------------|

**Remarks**

Currently supported tweakable parameters are as follows:

| Parameter name | Type    | Default value | Description  |
|----------------|---------|---------------|--|
| blinkSounds    | Boolean | true          | <p>Sound blinking is a sound shutdown for a short period of time (less than 1/100 of second) during slide switching. This technique is used only when the slide containing an embedded Flash movie is changed by some other slide. The reason for using this trick is that embedded Flash files can play sounds by their own and there is the only way to prevent possible sound issues. Sometimes this feature can be undesirable and can be turned off by setting the <b>blinkSound</b> parameter to false</p> |

**See also**

[IPlayer Interface](#)

[ISoundListener Interface](#)

[Flash Presentation ActionScript API Reference](#)

**1.5.2.31. fsplayer.api.ISoundListener Interface**

The **ISoundListener** interface defines the following set of methods invoked by sound controller when the corresponding events occur.

| Methods |             |
|---------|-------------|
| Method  | Description |
|         |             |

|   |   |
|---|---|
| onSoundVolumeChanged(soundController: <a href="#">ISoundCon</a> | This method is invoked when sound volume is changed. Use <code>getVolume()</code> method of the passed <a href="#">ISoundController</a> interface to get new volume |
|---|---|

In order to receive sound notifications you need to create a class which implements `ISoundListener` interface and pass it to the `addListener()` method of the [ISoundController](#) Interface.

### See also

[ISoundController Interface](#)

[Flash Presentation ActionScript API Reference](#)

### 1.5.2.32. fsplayer.api.IThumbnailsCollection Interface

The **IThumbnailsCollection** Interface represents a collection of slide thumbnails. It provides the following methods for retrieving thumbnail dimensions and accessing individual thumbnails.

| Methods   |  |
|---|--|
| Method  | Description  |
| <code>getThumbnail(slideIndex:Number): <a href="#">ISlideThumbnail</a></code> | Returns the <a href="#">ISlideThumbnail</a> interface representing the particular presentation thumbnail within thumbnails collection  |
| <code>getThumbnailWidth():Number</code>                                       | Returns thumbnail width in pixels  |
| <code>getThumbnailHeight():Number</code>                                      | Returns thumbnail height in pixels   |
| <code>getThumbnailsCount():Number</code>                                      | Returns the number of thumbnails in the presentation. <b>Note:</b> The number of thumbnails is the same as the number of slides within the presentation. If the presentation skin doesn't support thumbnails or if the thumbnails were turned off in skin settings, the number of thumbnails is equal to 0 |

### Remarks

The **IThumbnailsCollection** interface is mainly used for the development of custom skins supporting thumbnails.

### See also

[ISlideThumbnail Interface](#)

[IPresentationInfo Interface](#)

[Flash Presentation ActionScript API Reference](#)

### 1.5.2.33. ActionScript 3 and Flex 2 support

#### iSpring.as2player.AcquireFocusEvent Class

The **AcquireFocusEvent** Class extends the standard [flash.events.Event](#) class and exposes the following constants and properties:

| Constants         |        |                         |   |
|-------------------|--------|-------------------------|---|
| Constant          | Type   | Value                   | Description   |
| KEYBOARD_FOCUS_ST | String | keyboardFocusStateChang | Defines the value of the <b>type</b> property of a <b>keyboardFocusStateChanged</b> event object. This type of event occurs when interactive element of the presentation (e.g. a text field of a Quiz) acquires or loses keyboard focus. A <b>keyboardFocusStateChanged</b> event contains the following properties: <a href="#">acquireFocus</a> . |

| Properties   |         |   |
|--------------|---------|---|
| Property     | Type    | Description   |
| acquireFocus | Boolean | Indicates whether the interactive element of the presentation acquires or lost keyboard focus. The following event contain this property: <a href="#">keyboardFocusStateChanged</a> |

#### See also

[IPresentationPlaybackController Interface](#)

## [ActionScript3 and Flex support in iSpring](#)

### iSpring.as2player.IAnimationStep Interface

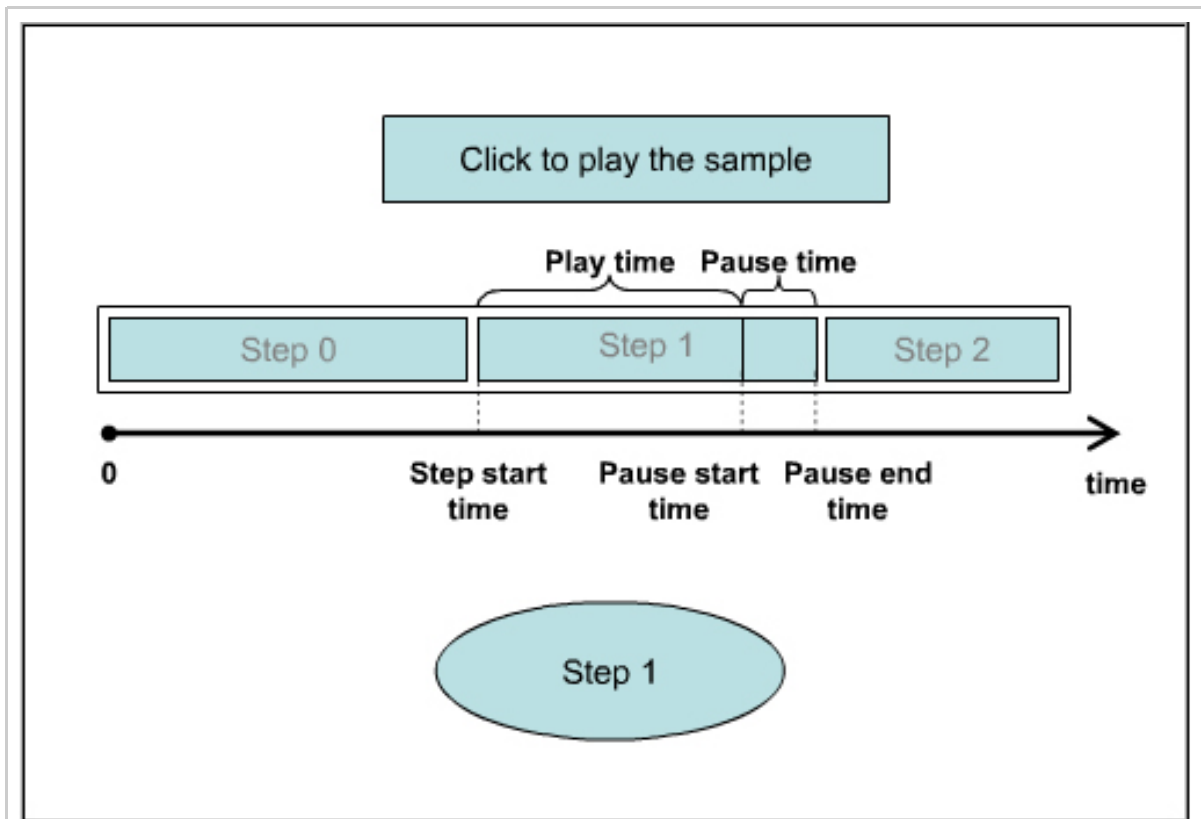
The **IAnimationStep** interface provides the information about animation step timings. It exposes the following properties:

| Properties     |        |   |
|----------------|--------|---|
| Property       | Type   | Description   |
| playTime       | Number | Animation step play time (see the <a href="#">remarks</a> section)        |
| pauseTime      | Number | Animation step pause time (see the <a href="#">remarks</a> section)       |
| startTime      | Number | Animation step start time (see the <a href="#">remarks</a> section)       |
| pauseStartTime | Number | Animation step pause start time (see the <a href="#">remarks</a> section) |
| pauseEndTime   | Number | Animation step pause end time (see the <a href="#">remarks</a> section)   |

#### Remarks

Each slide consists of one or more animation steps with the following structure.

|  |
|--|
|  |
|--|



#### Animation step structure

An Animation step starts playing at the **Step start time**. Animations are playing for the **Play time**. Then, at the **Pause start time**, animations are suspended for the **Pause time** period till the **Pause end time**. Pause time depends on slide timings that were set using "Record narration" or "Rehearse timings" menu of PowerPoint.

```
stepStartTime0 = 0
pauseStartTimei = stepStartTimei + playTimei
pauseEndTimei = pauseStartTimei + pauseTimei
stepStartTimei+1 = pauseEndTimei
```

If the slide has no timings (i.e. user has to click slide background to advance mouse-driven animations), **Pause time** and **Pause end time** are undefined. In this case returned timings are as follows:

```
pauseStartTimei = stepStartTimei + playTimei
pauseEndTimei = undefined
```

```
stepStartTime+1 = pauseStartTime
```

**See also**

[IAnimationSteps Interface](#)  
[ActionScript 3 and Flex support in iSpring](#)

**iSpring.as2player.IAnimationSteps Interface**

The **IAnimationSteps** Interface provides the information about animation steps on the slide.

| Properties |        |   |
|------------|--------|---|
| Property   | Type   | Description   |
| duration   | Number | The total duration of animation steps in seconds. The duration of animation steps is equal to the duration of the slide |
| stepsCount | Number | Number of slide animation steps   |

| Methods   |   |
|---|---|
| Method  | Description   |
| getStep(index:Number): <a href="#">IAnimationStep</a> | Returns the <a href="#">IAnimationStep</a> Interface providing the information about animation step timings |

**Remarks**

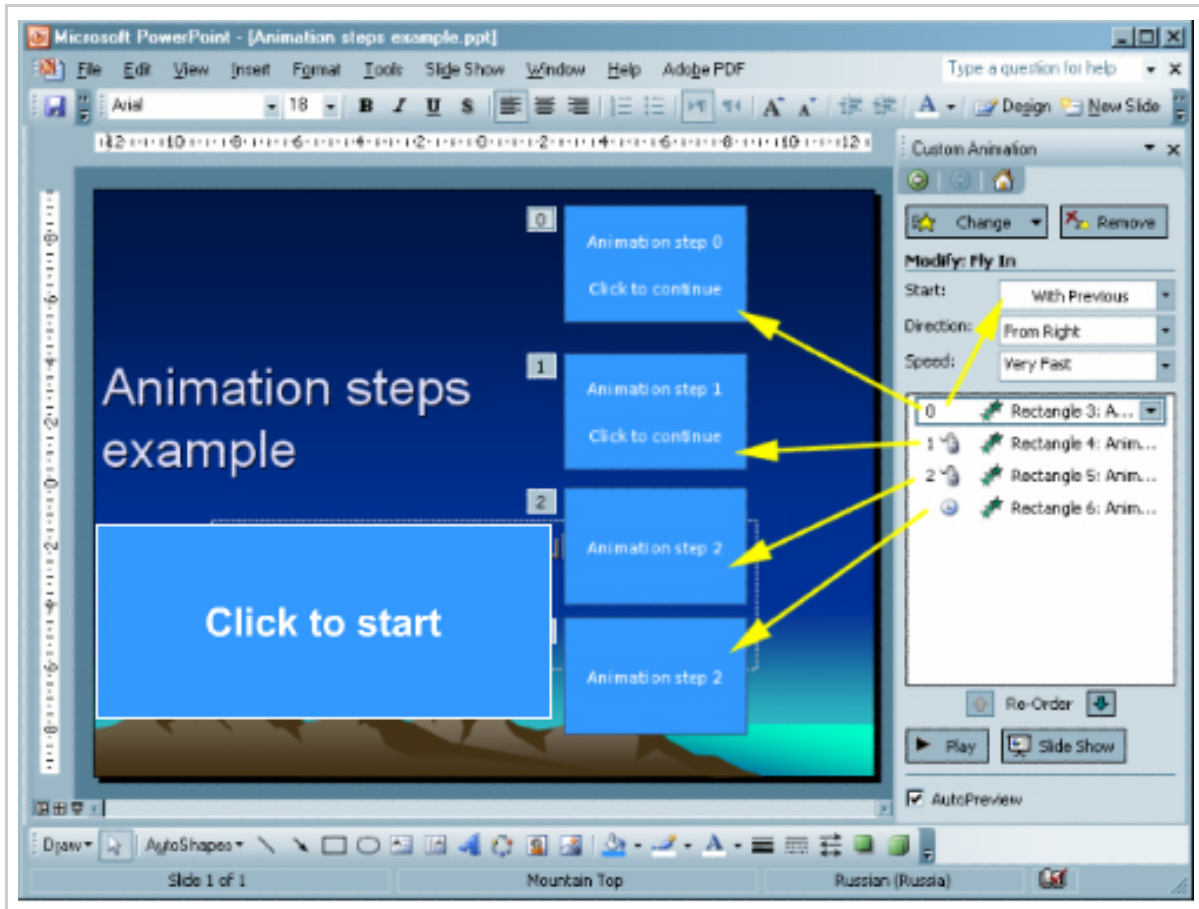
Slide animation timeline consists of one or more animation steps. Animation step is a sequence of animation effects playing without mouse click requirement.

Slide background, static shapes, and shapes animated before the first mouse driven animation are playing at the animation step 0. So there is at least 1 animation step in each slide.

Each mouse-driven animation and the subsequent non mouse-driven animations are playing at the separate animation step. In other words, the number of animation steps is equal to number of on-click animations plus 1.

**Example**

There are 4 animation effects in the following slide. They are represented by 3 animation steps in the generated Flash presentation (click to start animations).



- Effect 1 starts "With previous" animation effect. It is played at animation step 0
- Effects 2 and 3 start "On Click". They are played at animation steps 1 and 2
- Effect 4 starts "After previous" animation effect. It is played at animation step 2 after Effect 3.

### See also

[IAnimationStep Interface](#)

[SlideInfo Interface](#)

[ActionScript 3 and Flex support in iSpring](#)

**iSpring.as2player.ICompanyInfo Interface**

The **ICompanyInfo** Interface provides the following methods for retrieving company information.

| Properties |        |  |
|------------|--------|--|
| Property   | Type   | Description  |
| logo       | Number | Returns the <a href="#">ICompanyLogo</a> interface providing information about company logotype. |

| Methods                         |   |
|---------------------------------|---|
| Method                          | Description                             |
| hasLogo()(index:Number):Boolean | Indicates whether the company has logo. |

#### See also

[ICompanyLogo Interface](#)

[IPresentationInfo Interface](#)

[ActionScript 3 and Flex support in iSpring](#)

#### iSpring.as2player.ICompanyLogo Interface

The **ICompanyLogo** Interface provides the following methods for retrieving company information.

| Properties      |        |  |
|-----------------|--------|--|
| Property        | Type   | Description  |
| hyperlinkUrl    | String | Returns a web page URL, for example, the company web site address, which should be opened when user clicks the company logo. <b>Note:</b> An empty string indicates that no hyperlink is attached to the company logo. |
| hyperlinkTarget | String | Returns the target window name ("_self", "_blank", "_parent", "_top", or custom window name) where company logo hyperlink  |

|  |  |                  |
|--|--|------------------|
|  |  | should be opened |
|--|--|------------------|

**See also**

[ICompanyLogo Interface](#)

[ActionScript 3 and Flex support in iSpring](#)

**iSpring.as2player.IPlayer Interface**

The **IPlayer** Interface extends the standard [flash.events.IEventDispatcher](#) interface and exposes the following properties and events:

| Properties         |   |   |
|--------------------|---|---|
| Property           | Type  | Description   |
| initialized        | Boolean   | Returns a Boolean value indicating whether the player was initialized. The following properties return an <b>undefined</b> value until the player gets initialized: <ul style="list-style-type: none"> <li>• presentationInfo</li> <li>• playbackController</li> <li>• soundController</li> <li>• settings</li> </ul> |
| playbackController | <a href="#">IPresentationPlaybackController</a> | Returns the <a href="#">IPresentationPlaybackController</a> interface providing presentation playback control and navigation facilities.  |
| soundController    | <a href="#">ISoundController</a>                | Returns the <a href="#">ISoundController</a> interface providing sound volume control facilities.   |
| presentationInfo   | <a href="#">IPresentationInfo</a>               | Returns the <a href="#">IPresentationInfo</a> interface providing the presentation content information.   |
| settings           | Object  | Returns an Object storing playback and navigation settings as well as skin related settings (see the <a href="#">remarks</a> section)   |

|  |  |  |
|--|--|--|
|  |  | This method is mainly used for the custom skins development. In most cases you don't need to use it. |
|--|--|--|

| Events                                  |   |
|---|---|
| Type                                    | Description   |
| <a href="#">PlayerEvent.PLAYER_INIT</a> | Occurs when the ActionScript player module is connected with another Flash movie and ready to receive commands. |

**Remarks**

An object returned by the **getSettings()** method contains a set of implementation specific properties. Currently, this object has the following structure:

- **navigation** (Object) - the set of navigation properties:
  - **keyboard** (Object) - keyboard navigation properties:
    - **enabled** (String) - determines whether keyboard navigation is enabled ("1") or not
  - **mouse** (Object) - mouse navigation properties
    - **enabled** (String) - determines whether navigation by mouse clicks on the slide background is enabled ("1") or not
    - **target** (String) - determines whether the mouse click navigation is performed by slides ("slide") or steps ("step")
- **colorScheme** (Object) - color scheme settings:
  - **primaryColor** (String) - this property is obsolete and must not be used in new skins
- **appearance** (Object) - appearance settings:
  - **fullScreen** (String) - determines whether player window is opened full screen ("1") or not
- **playback** (Object) - playback settings:
  - **autoStart** (String) - determines whether the player starts presentation playback automatically ("1" or undefined) or not
- **webLinks** (Object) - the set of external web links related to the presentation:
  - **slideAreaLink** (Object) - web link related to the slide area
    - **href** (String) - web page reference, e.g. "*http://www.ispringsolutions.com*"
    - **target** (String) - target window, e.g. "*\_self*", "*\_blank*", "*\_parent*" or *custom window name*

- **presentationEndLink** (Object) - web link related to the presentation finish event
  - **href** (String) - web page reference, e.g. "<http://www.ispringsolutions.com>"
  - **target** (String) - target window, e.g. "\_self", "\_blank", "\_parent" or custom window name

### See also

[IPresentationPlaybackController Interface](#)

[PresentationInfo Interface](#)

[ISoundController Interface](#)

[flash.events.IEventDispatcher Interface](#)

[ActionScript 3 and Flex support in iSpring](#)

### iSpring.as2player.IPresentationInfo Interface

The **IPresentationInfo** class stores the presentation information and provides the following properties:

| Properties    |                                   |  |
|---------------|-----------------------------------|--|
| Property      | Type                              | Description  |
| title         | String                            | Presentation title   |
| slides        | <a href="#">ISlidesCollection</a> | Returns the <a href="#">ISlidesCollection</a> Interface providing information about presentation slides.   |
| slideWidth    | Number                            | Presentation slide width in pixels   |
| slideHeight   | Number                            | Presentation slide height in pixels  |
| presenterInfo | <a href="#">IPresenterInfo</a>    | Returns the <a href="#">IPresenterInfo</a> Interface providing the information about presentation presenter.   |
| frameRate     | Number                            | Presentation frame rate (number of frames per second)<br><b>Note:</b> when iSpring generated presentation is loaded by another Flash movie, it always plays at its |

|                 |                                       |  |
|-----------------|---------------------------------------|--|
|                 |                                       | frame rate.  |
| duration        | Number                                | Presentation duration in seconds   |
| visibleDuration | Number                                | Visible slides duration in seconds   |
| companyInfo     | <a href="#">ICompanyInfo</a>          | Returns the <a href="#">ICompanyInfo</a> Interface providing the company information.                                |
| presenters      | <a href="#">IPresentersCollection</a> | Returns the <a href="#">IPresentersCollection</a> Interface providing the information about presentation presenters. |
| references      | <a href="#">IReferencesCollection</a> | Returns the <a href="#">IReferencesCollection</a> Interface providing the information about presentation references. |

| Methods                  |  |
|--------------------------|--|
| Method                   | Description  |
| hasCompanyInfo():Boolean | Indicates whether the presentation contains the company information. |
| hasPresenter():Boolean   | Indicates whether the presentation contains presenter information.   |
| hasReferences():Boolean  | Indicates whether the presentation contains references               |

**See also**

- [CompanyInfo Interface](#)
- [PresenterInfo Interface](#)
- [IPresentersCollection Interface](#)
- [SlidesCollection Interface](#)
- [ReferencesCollection Interface](#)
- [ActionScript 3 and Flex support in iSpring](#)

**iSpring.as2player.IPresentationPlaybackController Interface**

The **IPresentationPlaybackController** interface extends the standard [flash.events.IEventDispatcher](#) Interface and provides the following properties, methods and events for presentation playback and navigation control.

| Properties                   |         |   |
|------------------------------|---------|---|
| Property                     | Type    | Description   |
| playing                      | Boolean | Indicates whether the presentation is playing or not (read only)  |
| currentSlideDuration         | Number  | A duration of the currently playing slide (read only)   |
| currentSlideIndex            | Number  | Current slide index (read only)   |
| currentSlidePlaybackPosition | Number  | Returns playback position within current slide.<br><b>Note:</b> playback position is a number between 0 and 1   |
| currentStepIndex             | Number  | Returns current animation step index  |
| animationStepPause           | Number  | Sets a pause between animation steps. If an <b>undefined</b> value is passed as pause value, animation steps will not be played automatically.  |
| automaticSlideSwitching      | Boolean | Enables or disables automatic switching to the next slide.<br><b>Note:</b> this property affects automatically switching slides only.<br>Returns true if automatic slide switching has been enabled, false otherwise.<br><b>Note:</b> automatic slide switching is enabled by default |
|                              |         |   |

|                          |        |   |
|--------------------------|--------|---|
| currentVisibleSlideIndex | Number | Current visible slide index (read only) |
| nextSlideIndex           | Number | Next slide index (read only)            |
| previousSlideIndex       | Number | Previous slide index (read only)        |

| Methods   |  |
|---|--|
| Method  | Description  |
| play():Void   | Resumes presentation playback  |
| pause():Void  | Suspends presentation playback   |
| gotoNextSlide([autoStart:Boolean = true]):Void                | Switches to the next presentation slide and starts its playback.<br><b>Note:</b> if the autoStart parameter was set to false, slide playback would not be started after switching  |
| gotoPreviousSlide([autoStart:Boolean = true]):Void            | Switches to the previous presentation slide and starts its playback.<br><b>Note:</b> if the autoStart parameter was set to false, slide playback would not be started after switching  |
| gotoLastViewedSlide([autoStart:Boolean = true]):Void          | Switches to the previously viewed slide and starts its playback.<br><b>Note:</b> if the <b>autoStart</b> parameter was set to false, slide playback would not be started after switching                                       |
| gotoSlide(slideIndex:Number, [autoStart:Boolean = true]):Void | Switches to the given slide of the presentation and starts its playback<br><b>Notes:</b> Slide index is zero-based.<br>If the <b>autoStart</b> parameter was set to false, slide playback would not be started after switching |
| pauseCurrentSlideAt(position:Number):Void                     | Suspends currently playing slide at the given position<br><b>Note:</b> position is a number in between 0 and 1   |
| playCurrentSlideFrom(position:Number):Void                    | Restarts current slide from the given position.<br><b>Note:</b> position is a number between 0 and 1   |

|   |  |
|---|--|
| seek(position:Number):Void  | Seeks the current slide to the given position. After this operation slide playback gets suspended until the <b>endSeek()</b> method or other playback control method is called<br><b>Note:</b> position is a number between 0 and 1            |
| endSeek([resumePlayback:Boolean = undefined]):Void                          | Exits from seeking mode and restores playback.<br><b>Note:</b> the <b>resumePlayback</b> parameter allows to start or pause slide after seeking explicitly   |
| gotoNextStep():Void   | Plays current slide from the beginning of the next animation step.<br><b>Note:</b> being called at the end of the slide this method plays presentation from the beginning of the next slide  |
| gotoPreviousStep():Void   | Moves slide playback one step back. This method suspends slide playback at the end of the previous animation step.<br><b>Note:</b> being called on the start slide step this method causes playback switching to the end of the previous slide |
| playFromStep(stepIndex:Number):Void   | Plays current slide from the beginning of the specified animation step   |
| pauseAtStepStart(stepIndex:Number):Void                                     | Suspends current slide at the beginning of the specified animation step  |
| pauseAtStepEnd(stepIndex:Number):Void                                       | Suspends current slide at the end of specified animations step   |
| gotoVisibleSlide(visibleSlideIndex:Number, [autoStart:Boolean = true]):Void | Switches to the given slide within visible slides and starts its playback.<br><b>Notes:</b> Slide index is zero-based.<br>If the <b>autoStart</b> parameter was set to false, slide playback would not be started after switching.             |
| gotoFirstSlide([autoStart:Boolean = true]):Void                             | Switches to the first presentation slide and starts its playback.<br><b>Notes:</b> Slide index is zero-based.<br>If the <b>autoStart</b> parameter was set to false, slide playback would not be started after switching.                      |

|   |  |
|---|--|
| <code>gotoLastSlide([autoStart:Boolean = true]):Void</code> | Switches to the last presentation slide and starts its playback.<br><b>Notes:</b> Slide index is zero-based.<br>If the <b>autoStart</b> parameter was set to false, slide playback would not be started after switching. |
|---|--|

| Events   |   |
|--|---|
| Type   | Description   |
| <a href="#">SlidePlaybackEvent.SLIDE_LOADING_COMPLETE</a>      | Occurs when a slide of the presentation is loaded.  |
| <a href="#">SlidePlaybackEvent.CURRENT_SLIDE_INDEX_CHANGED</a> | Occurs when current slide is changed by another slide of the presentation.  |
| <a href="#">StepPlaybackEvent.ANIMATION_STEP_CHANGED</a>       | Occurs when currently playing animation step index is changed.  |
| <a href="#">PlaybackEvent.START_PLAYBACK</a>                   | Occurs when presentation playback is started.   |
| <a href="#">PlaybackEvent.PAUSE_PLAYBACK</a>                   | Occurs when presentation playback is suspended.   |
| <a href="#">PlaybackEvent.PRESENTATION_PLAYBACK_COMPLETE</a>   | Occurs when presentation playback is complete.  |
| <a href="#">PlaybackEvent.SEEKING_COMPLETE</a>                 | Occurs when a seek command execution is completed.  |
| <a href="#">PlaybackPositionEvent.SLIDE_POSITION_CHANGED</a>   | Occurs when the playback position of the currently playing slide is changed.  |
| <a href="#">AcquireFocusEvent.KEYBOARD_FOCUS_STATE_CHANGED</a> | Occurs when interactive element of the presentation (e.g. a text field of a Quiz) acquires or loses keyboard focus. |

**See also**[IPlayer Interface](#)[SlidePlaybackEvent Class](#)[PlaybackEvent Class](#)[StepPlaybackEvent Class](#)

[PlaybackPositionEvent Class](#)  
[AcquireFocusEvent Class](#)  
[ActionScript 3 and Flex support in iSpring](#)

### iSpring.as2player.IPresenterInfo Interface

The **IPresenterInfo** Interface stores the information about presenter and provides the following properties:

| Properties    |        |                                     |
|---------------|--------|-------------------------------------|
| Property      | Type   | Description                         |
| name          | String | Presenter name                      |
| title         | String | Presenter title                     |
| biographyText | String | A brief information about presenter |
| email         | String | Presenter e-mail address            |
| webSite       | String | Presenter Web site address          |
| index         | Number | Index within presenters collection  |

| Methods            |  |
|--------------------|--|
| Method             | Description                                |
| hasPhoto():Boolean | Indicates whether the presenter has photo. |

### See also

[IPresentationInfo Interface](#)  
[ActionScript 3 and Flex support in iSpring](#)

### iSpring.as2player.IPresentersCollection Interface

The **IPresentersCollection** Interface provides the following methods for retrieving the presenters information:

| Properties |        |  |
|------------|--------|--|
| Property   | Type   | Description                            |
| count      | Number | Number of presenters in the collection |

| Methods  |  |
|--|--|
| Method   | Description  |
| getPresenter(index:Number): <a href="#">IPresenterInfo</a> | Returns <a href="#">IPresenterInfo</a> interface providing the information about the specified presenter. Note: the index parameter is an integer value from 0 to (number of presenters - 1) |

**See also**

- [IPresenterInfo Interface](#)
- [IPresentationInfo Interface](#)
- [ActionScript 3 and Flex support in iSpring](#)

**iSpring.as2player.IPresenterVideo Interface**

The **IPresenterVideo** interface provides the following methods for presenter video information retrieving:

| Properties |        |                                |
|------------|--------|--------------------------------|
| Property   | Type   | Description                    |
| width      | Number | Width of the presenter video.  |
| height     | Number | Height of the presenter video. |

**See also**

- [IPresenterInfo Interface](#)
- [ActionScript 3 and Flex support in iSpring](#)

**iSpring.as2player.IReferenceInfo Interface**

The **IReferenceInfo** Interface provides the following properties for retrieving reference

information:

| Properties |        |   |
|------------|--------|---|
| Property   | Type   | Description   |
| title      | String | Reference title   |
| url        | String | Reference URL   |
| target     | String | Target window name where the reference should be opened |

**See also**

[IReferencesCollection Interface](#)  
[ActionScript 3 and Flex support in iSpring](#)

#### **iSpring.as2player.IReferencesCollection Interface**

The **IReferencesCollection** interface provides the following properties and methods for accessing presentation references:

| Properties |        |   |
|------------|--------|---|
| Property   | Type   | Description                             |
| count      | Number | Number of references in the collection. |

| Methods  |   |
|--|---|
| Method   | Description   |
| getReference(index:Number): <a href="#">IReferenceInfo</a> | Returns <a href="#">ReferenceInfo</a> interface providing the information about the specified reference.<br><b>Note:</b> the <b>index</b> parameter is an integer value from 0 to (number of references - 1). |

**See also**

[IReferenceInfo Interface](#)  
[IPresentationInfo Interface](#)

## [ActionScript 3 and Flex support in iSpring](#)

### **iSpring.as2bridge.ISlideInfo Interface**

The **ISlideInfo** Interface provides information about the particular slide of the presentation such as title, duration, text, etc. It exposes the following properties and methods:

| Properties       |                                 |   |
|------------------|---------------------------------|---|
| Property         | Type                            | Description   |
| loaded           | Boolean                         | Indicating whether the slide has been loaded completely.  |
| duration         | Number                          | Slide duration in seconds.  |
| startTime        | Number                          | Slide start time within presentation timeline.  |
| endTime          | Number                          | Slide end time within presentation timeline.  |
| level            | Number                          | Slide level (0 - upper level).  |
| hidden           | Boolean                         | Indicates whether the slide is hidden.  |
| index            | Number                          | Slide index within presentation.  |
| visibleIndex     | Number                          | Slide index within visible slides.  |
| visibleStartTime | Number                          | Slide start time within visible slides timeline.  |
| visibleEndTime   | Number                          | Slide end time within visible slides timeline.  |
| title            | String                          | Slide title   |
| animationSteps   | <a href="#">IAnimationSteps</a> | Returns the <a href="#">IAnimationSteps</a> interface providing the information about slide animation steps |

|                       |                                  |  |
|-----------------------|----------------------------------|--|
| notesText             | String                           | Slide notes text   |
| startStepIndex        | Number                           | Slide start step index within the presentation   |
| endStepIndex          | Number                           | Slide end step index within the presentation   |
| slideText             | String                           | Slide text in the normalized form (see the <a href="#">remarks</a> section)  |
| notesTextNormalized   | String                           | Slide notes text in the <b>normalized form</b> (see the <a href="#">remarks</a> section)   |
| titleNormalized       | String                           | Slide title in the <b>normalized form</b> (see the <a href="#">remarks</a> section)  |
| visibleStartStepIndex | Number                           | Slide start step index within visible slides   |
| visibleEndStepIndex   | Number                           | Slide end step index within visible slides   |
| presenterIndex        | Number                           | Index of presenter associated with the slide <b>Note:</b> This method returns an undefined value if no presenter is associated with the slide. |
| resources             | <a href="#">ISlidesResources</a> | Returns the <a href="#">ISlidesResources</a> interface providing the information about resources included into the slide.                      |

#### Remarks

Normalized strings are strings having all characters in the lower case with no extra spaces between words. This simplifies and speeds up full text search implementation.

#### See also

[IAnimationSteps Interface](#)  
[ISlidesCollection Interface](#)  
[ActionScript 3 and Flex support in iSpring](#)

#### **iSpring.as2bridge.ISlideResources Interface**

The **ISlideResources** Interface provides information about the particular slide of the presentation such as title, duration, text, etc. It exposes the following properties and methods:

| Properties     |                                 |  |
|----------------|---------------------------------|--|
| Property       | Type                            | Description  |
| presenterVideo | <a href="#">IPresenterVideo</a> | Returns the <a href="#">IPresenterVideo</a> interface providing information about presenter video. If there is no presenter video, an undefined value is returned. |

| Methods                     |   |
|-----------------------------|---|
| Method                      | Description                                       |
| hasPresenterVideo():Boolean | Indicate whether the slide has a presenter video. |

#### **See also**

[IPresenterVideo Interface](#)  
[ISlideInfo Interface](#)  
[ActionScript 3 and Flex support in iSpring](#)

#### **iSpring.as2player.ISlidesCollection Interface**

The **ISlidesCollection** interface represents a collection of presentation slides. It provides the following properties and methods:

| Properties  |        |                                      |
|-------------|--------|--------------------------------------|
| Property    | Type   | Description                          |
| slidesCount | Number | Number of slides in the presentation |

|                    |        |  |
|--------------------|--------|--|
| visibleSlidesCount | Number | Number of visible slides in the presentation |
|--------------------|--------|--|

| Methods   |  |
|---|--|
| Method  | Description  |
| getSlideInfo(slideIndex:Number): <a href="#">ISlideInfo</a>           | Returns <a href="#">SlideInfo</a> Interface providing information about the specified slide<br><b>Note:</b> Slide index is zero-based.                 |
| getVisibleSlide(visibleSlideIndex:Number): <a href="#">ISlideInfo</a> | Returns <a href="#">SlideInfo</a> Interface providing information about the specified visible slide<br><b>Note:</b> Visible slide index is zero-based. |

**See also**[IPresentationInfo Interface](#)[ISlideInfo Interface](#)[ActionScript 3 and Flex support in iSpring](#)**iSpring.as2player.ISoundController Interface**

The **ISoundController** interface provides sound volume control facilities. It extends the standard [flash.events.IEventDispatcher](#) interface and provides the following properties and events.

| Properties |        |   |
|------------|--------|---|
| Property   | Type   | Description   |
| volume     | Number | Sound volume<br><b>Note:</b> sound volume is a number between 0 and 1 |

| Events  |   |
|---|---|
| Type  | Description                             |
| <a href="#">SoundEvent.SOUND_VOLUME_CHANGED</a> | Occurs when the sound volume is changed |

**See also**

[IPlayer Interface](#)[flash.events.IEventDispatcher Interface](#)[ActionScript 3 and Flex support in iSpring](#)**iSpring.as2player.PlaybackEvent Class**

The **PlaybackEvent** Class extends the standard [flash.events.Event](#) class and exposes the following constants and properties:

| Constants                      |        |                              |  |
|--------------------------------|--------|------------------------------|--|
| Constant                       | Type   | Value                        | Description  |
| PAUSE_PLAYBACK                 | String | pausePlayback                | Defines the value of the <b>type</b> property of a <b>pausePlayback</b> event object. This type of event occurs when presentation playback is suspended.                             |
| START_PLAYBACK                 | String | startPlayback                | Defines the value of the <b>type</b> property of a <b>startPlayback</b> event object. This type of event occurs when presentation playback is resumed.                               |
| SEEKING_COMPLETE               | String | seekingComplete              | Defines the value of the <b>type</b> property of the <b>seekingComplete</b> event object. This type of event occurs when the seek command execution is completed.                    |
| PRESENTATION_PLAYBACK_COMPLETE | String | presentationPlaybackComplete | Defines the value of the <b>type</b> property of the <b>presentationPlaybackComplete</b> event object. This type of event occurs when the playback of the presentation is completed. |

**See also**

[IPresentationPlaybackController Interface](#)

[IPlayer Interface](#)

[ActionScript3 and Flex support in iSpring](#)

**iSpring.as2player.PlaybackPositionEvent Class**

The **PlaybackPositionEvent** Class extends the standard [ispring.as2player.PlaybackEvent](#) class and exposes the following constants and properties:

| Constants              |        |                      |   |
|------------------------|--------|----------------------|---|
| Constant               | Type   | Value                | Description   |
| SLIDE_POSITION_CHANGED | String | slidePositionChanged | Defines the value of the <b>type</b> property of the <b>slidePositionChanged</b> event object. This type of event occurs when the playback position of the currently playing slide is changed. A slidePositionChanged event contains the following properties: <a href="#">position</a> . |

| Properties |        |   |
|------------|--------|---|
| Property   | Type   | Description   |
| position   | Number | Slide playback position. The following events contain this property: <a href="#">slidePositionChanged</a> |

**See also**

[IPresentationPlaybackController Interface](#)

[ActionScript3 and Flex support in iSpring](#)

**iSpring.as2player.PlayerEvent Class**

The **PlayerEvent** Class extends the standard [flash.events.Event](#) class and exposes the following constants and properties:

| Constants   |        |            |   |
|-------------|--------|------------|---|
| Constant    | Type   | Value      | Description   |
| PLAYER_INIT | String | playerInit | Defines the value of the <b>type</b> property of a <b>playerInit</b> event object. This type of event occurs when player initialization is completed. |

### See also

[IPlayer Interface](#)

[ActionScript3 and Flex support in iSpring](#)

### iSpring.as2player.PresentationLoader Class

The **PresentationLoader** Class extends the standard [flash.display.Sprite](#) class and exposes the following methods and events:

| Properties        |   |   |
|-------------------|---|---|
| Property          | Type  | Description                                     |
| content           | <a href="#">flash.display.DisplayObject</a> | Content object which contains the presentation. |
| contentLoaderInfo | LoaderInfo                                  | Content loader info                             |
| player            | <a href="#">IPlayer</a>                     | Presentation player                             |

| Methods              |   |
|----------------------|---|
| Method               | Description   |
| PresentationLoader() | Constructs a new PresentationLoader object which creates player and establishes local connection with the iSpring generated |

|   |   |
|---|---|
|   | Flash presentation via ActionScript Connector.                                      |
| load(request:URLRequest, securityDomain:SecurityDomain = null):void | Loads the iSpring generated Flash presentation specified in request parameter.      |
| unload():void   | Unloads the iSpring generated Flash presentation and closes ActionScript Connector. |
| close():void  | Closes the presentation loader.   |

**See also**

[ActionScript 3 and Flex support in iSpring](#)

**iSpring.as2player.SlidePlaybackEvent Class**

The **SlidePlaybackEvent** Class extends the standard [ispring.as2player.PlaybackEvent](#) class and exposes the following constants and properties:

| Constants          |        |                          |  |
|--------------------|--------|--------------------------|--|
| Constant           | Type   | Value                    | Description  |
| CURRENT_SLIDE_INDE | String | currentSlideIndexChanged | Defines the value of the <b>type</b> property of the <b>currentSlideIndexChanged</b> event object. This type of event occurs when current slide is changed by another slide of the presentation. A <b>currentSlideIndexChanged</b> event contains the following properties: <a href="#">slideIndex</a> . |
| SLIDE_LOADING_COM  | String | slideLoadingComplete     | Defines the value of the <b>type</b> property of the <b>slideLoadingComplete</b> event object. This type of event occurs when the given slide of the presentation is loaded.   |

|  |  |  |  |
|--|--|--|--|
|  |  |  | A slideLoadingComplete event contains the following properties: <a href="#">slideIndex</a> . |
|--|--|--|--|

| Properties |        |   |
|------------|--------|---|
| Property   | Type   | Description   |
| slideIndex | Number | Slide index.<br>The following events contain this property:<br><a href="#">currentSlideIndexChanged</a> ,<br><a href="#">slideLoadingComplete</a> |

**See also**

[IPresentationPlaybackController Interface](#)  
[ActionScript3 and Flex support in iSpring](#)

**iSpring.as2player.SoundEvent Class**

The **SoundEvent** Class extends the standard [flash.events.Event](#) class and exposes the following constants and properties:

| Constants        |        |                    |   |
|------------------|--------|--------------------|---|
| Constant         | Type   | Value              | Description   |
| SOUND_VOLUME_CHA | String | soundVolumeChanged | Defines the value of the <b>type</b> property of the <b>soundVolumeChanged</b> event object. This type of event occurs when the sound volume is changed.<br>A soundVolumeChanged event contains the following properties:<br><a href="#">volume</a> . |

| Properties |      |             |
|------------|------|-------------|
| Property   | Type | Description |
|            |      |             |

|        |        |  |
|--------|--------|--|
| volume | Number | Sound volume.<br>The following events contain this property:<br><a href="#">soundVolumeChanged</a> |
|--------|--------|--|

**See also**

[IPresentationPlaybackController Interface  
ActionScript3 and Flex support in iSpring](#)

**iSpring.as2player.StepPlaybackEvent Class**

The **StepPlaybackEvent** Class extends the standard [ispring.as2player.PlaybackEvent](#) class and exposes the following constants and properties:

| Constants              |        |                      |   |
|------------------------|--------|----------------------|---|
| Constant               | Type   | Value                | Description   |
| ANIMATION_STEP_CHANGED | String | animationStepChanged | Defines the value of the <b>type</b> property of the <b>animationStepChanged</b> event object. This type of event occurs when current slide is changed by another slide of the presentation.<br>A <code>animationStepChanged</code> event contains the following properties:<br><a href="#">stepIndex</a> . |

| Properties |        |  |
|------------|--------|--|
| Property   | Type   | Description  |
| stepIndex  | Number | Animation step index.<br>The following events contain this property:<br><a href="#">currentSlideIndexChanged</a> ,<br><a href="#">slideLoadingComplete</a> |

**See also**

[IPresentationPlaybackController Interface  
ActionScript3 and Flex support in iSpring](#)

**Obsolete ActionScript 3 API**

**iSpring.as3bridge.AnimationStep Class**

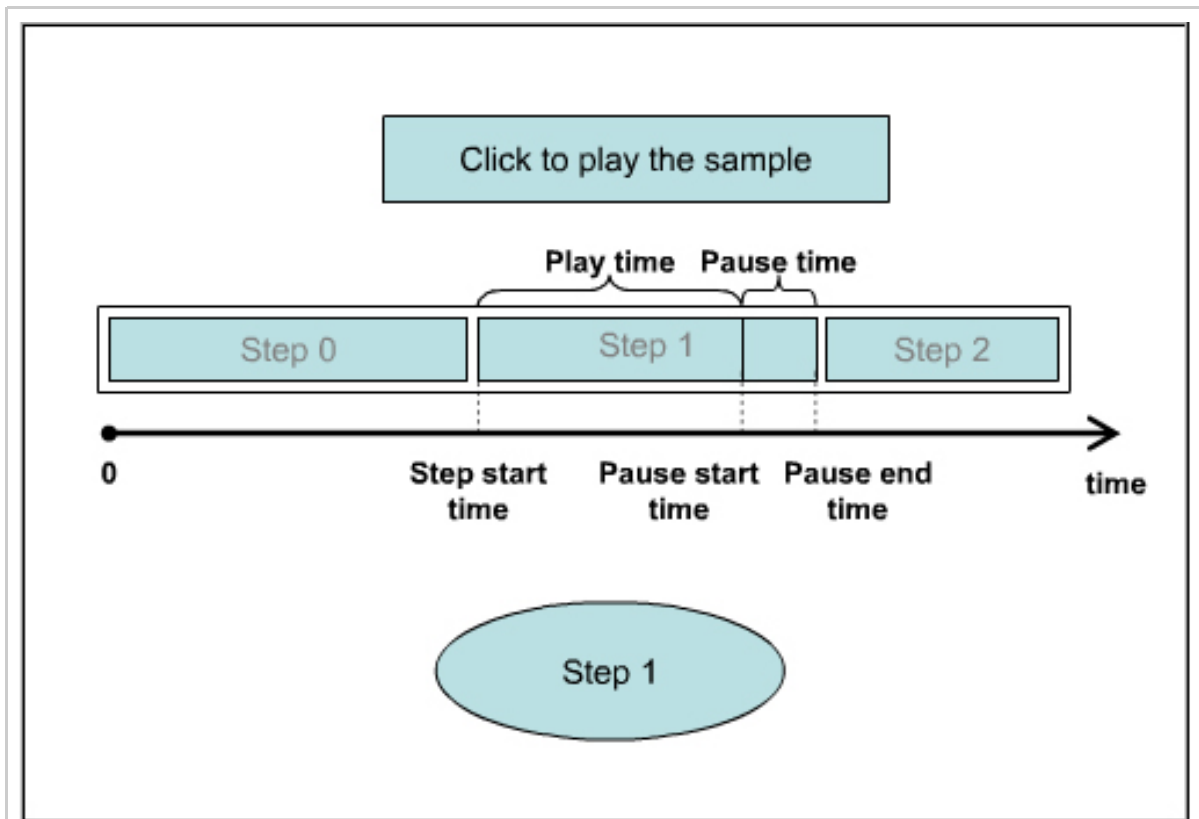
The **AnimationStep** class provides the information about animation step timings. It exposes the following properties:

| Properties     |        |   |
|----------------|--------|---|
| Property       | Type   | Description   |
| playTime       | Number | Animation step play time (see the <a href="#">remarks</a> section)        |
| pauseTime      | Number | Animation step pause time (see the <a href="#">remarks</a> section)       |
| startTime      | Number | Animation step start time (see the <a href="#">remarks</a> section)       |
| pauseStartTime | Number | Animation step pause start time (see the <a href="#">remarks</a> section) |
| pauseEndTime   | Number | Animation step pause end time (see the <a href="#">remarks</a> section)   |

**Remarks**

Each slide consists of one or more animation steps with the following structure.

|  |
|--|
|  |
|--|



#### Animation step structure

An Animation step starts playing at the **Step start time**. Animations are playing for the **Play time**. Then, at the **Pause start time**, animations are suspended for the **Pause time** period till the **Pause end time**. Pause time depends on slide timings that were set using "Record narration" or "Rehearse timings" menu of PowerPoint.

```
stepStartTime0 = 0
pauseStartTimei = stepStartTimei + playTimei
pauseEndTimei = pauseStartTimei + pauseTimei
stepStartTimei+1 = pauseEndTimei
```

If the slide has no timings (i.e. user has to click slide background to advance mouse-driven animations), **Pause time** and **Pause end time** are undefined. In this case returned timings are as follows:

```
pauseStartTimei = stepStartTimei + playTimei
pauseEndTimei = undefined
```

```
stepStartTimei+1 = pauseStartTimei
```

**See also**

[AnimationSteps Class](#)  
[Obsolete ActionScript 3 API](#)

**iSpring.as3bridge.AnimationSteps Class**

The **AnimationSteps** Class provides the information about animation steps on the slide.

| Properties |        |   |
|------------|--------|---|
| Property   | Type   | Description   |
| duration   | Number | The total duration of animation steps in seconds. The duration of animation steps is equal to the duration of the slide |
| stepsCount | Number | Number of slide animation steps   |

| Methods  |  |
|--|--|
| Method   | Description  |
| getStep(index:Number): <a href="#">AnimationStep</a> | Returns the <a href="#">AnimationStep</a> Class providing the information about animation step timings |

**Remarks**

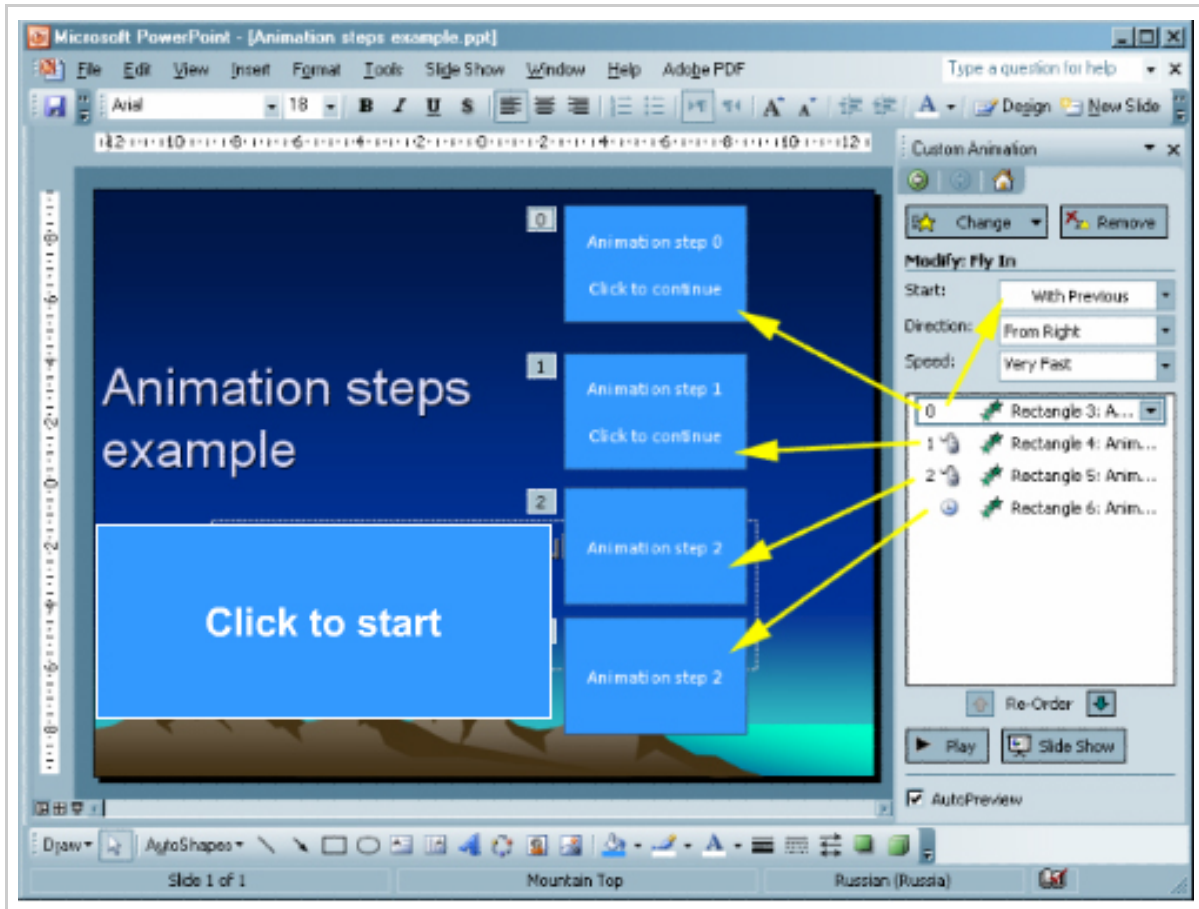
Slide animation timeline consists of one or more animation steps. Animation step is a sequence of animation effects playing without mouse click requirement.

Slide background, static shapes, and shapes animated before the first mouse driven animation are playing at the animation step 0. So there is at least 1 animation step in each slide.

Each mouse-driven animation and the subsequent non mouse-driven animations are playing at the separate animation step. In other words, the number of animation steps is equal to number of on-click animations plus 1.

**Example**

There are 4 animation effects in the following slide. They are represented by 3 animation steps in the generated Flash presentation (click to start animations).



- Effect 1 starts "With previous" animation effect. It is played at animation step 0
- Effects 2 and 3 start "On Click". They are played at animation steps 1 and 2
- Effect 4 starts "After previous" animation effect. It is played at animation step 2 after Effect 3.

### See also

[AnimationStep Class](#)

[SlideInfo Class](#)

[Obsolete ActionScript 3 API](#)

**iSpring.as3bridge.BridgeEvent Class**

The **BridgeEvent** Class extends the standard [flash.events.Event](#) class and exposes the following constants and properties:

| Constants       |        |                |   |
|-----------------|--------|----------------|---|
| Constant        | Type   | Value          | Description   |
| PAUSE           | String | pause          | Defines the value of the type property of a <b>pause</b> event object. This type of event occurs when presentation playback is suspended.<br>A pause event contains the following properties: <a href="#">playbackController</a>  |
| PLAY            | String | play           | Defines the value of the type property of a <b>play</b> event object. This type of event occurs when presentation playback is resumed.<br>A play event contains the following properties: <a href="#">playbackController</a>  |
| STEP_CHANGE     | String | stepChange     | Defines the value of the type property of the <b>stepChange</b> event object. This type of event occurs when currently playing animation step index is changed.<br>A stepChange event contains the following properties: <a href="#">playbackController</a> , <a href="#">stepIndex</a> |
| POSITION_CHANGE | String | positionChange | Defines the value of the type property of the <b>positionChange</b> event object. This type of event occurs when the playback position of the   |

|                     |        |                   |   |
|---------------------|--------|-------------------|---|
|                     |        |                   | <p>currently playing slide is changed.</p> <p>A positionChange event contains the following properties:</p> <p><a href="#">playbackController</a>,<br/><a href="#">position</a></p>   |
| SLIDE_CHANGE        | String | slideChange       | <p>Defines the value of the type property of the <b>slideChange</b> event object. This type of event occurs when current slide is changed by another slide of the presentation.</p> <p>A slideChange event contains the following properties:</p> <p><a href="#">playbackController</a>,<br/><a href="#">slideIndex</a></p> |
| SLIDE_LOAD_COMPLETE | String | slideLoadComplete | <p>Defines the value of the type property of the <b>slideLoadComplete</b> event object. This type of event occurs when the given slide of the presentation is loaded.</p> <p>A slideLoadComplete event contains the following properties:</p> <p><a href="#">playbackController</a>,<br/><a href="#">slideIndex</a></p>     |
| PLAYBACK_COMPLETE   | String | playbackComplete  | <p>Defines the value of the type property of the <b>playbackComplete</b> event object. This type of event occurs when the playback of the presentation is completed.</p> <p>A playbackComplete event contains the following properties:</p> <p><a href="#">playbackController</a></p>                                       |

|                  |        |                 |  |
|------------------|--------|-----------------|--|
| BRIDGE_LOADED    | String | bridgeLoaded    | Defines the value of the type property of the <b>bridgeLoaded</b> event object. This type of event occurs when the ActionScript bridge module is loaded and ready to receive commands from the parent Flash movie. A bridgeLoaded event contains the following properties:<br><a href="#">succeeded</a> , <a href="#">player</a> |
| PLAYER_INIT      | String | playerInit      | Defines the value of the type property of the <b>playerInit</b> event object. This type of event occurs when player initialization is completed, just after the <a href="#">bridgeLoaded</a> event. A playerInit event contains the following properties:<br><a href="#">succeeded</a> , <a href="#">player</a>                  |
| VOLUME_CHANGE    | String | volumeChange    | Defines the value of the type property of the <b>volumeChange</b> event object. This type of event occurs when the sound volume is changed. A volumeChange event contains the following properties:<br><a href="#">soundController</a> , <a href="#">volume</a>  |
| SEEKING_COMPLETE | String | seekingComplete | Defines the value of the type property of the <b>seekingComplete</b> event object. This type of event occurs when the seek command execution is completed. A seekingComplete event contains the following  |

|                          |        |                        |   |
|--------------------------|--------|------------------------|---|
|                          |        |                        | properties:<br><a href="#">playbackController</a> ,<br><a href="#">position</a>   |
| VOLUME_CHANGING_COMPLETE | String | volumeChangingComplete | Defines the value of the type property of the <b>volumeChangingComplete</b> event object. This type of event occurs when the sound volume property of the <a href="#">SoundController</a> object is changed.<br>A volumeChangingComplete event contains the following properties:<br><a href="#">soundController</a> , <a href="#">volume</a> |
| SLIDE_METADATA_LOADED    | String | slideMetadataLoad      | Defines the value of the type property of the <b>slideMetadataLoad</b> event object. This type of event is generated by <a href="#">SlideInfo</a> object when its metadata is loaded.<br>A slideMetadataLoad event contains the following properties:<br><a href="#">slideInfo</a> , <a href="#">slideIndex</a>                               |

| Properties         |                                    |  |
|--------------------|------------------------------------|--|
| Property           | Type                               | Description  |
| playbackController | <a href="#">PlaybackController</a> | The <a href="#">PlaybackController</a> class instance.<br>The following events contain this property:<br><a href="#">play</a> , <a href="#">pause</a> , <a href="#">stepChange</a> ,<br><a href="#">positionChange</a> , <a href="#">slideChange</a> ,<br><a href="#">slideLoadComplete</a> ,<br><a href="#">playbackComplete</a> ,<br><a href="#">seekingComplete</a> |
| slideIndex         | Number                             | Slide index.   |

|                 |                                 |   |
|-----------------|---------------------------------|---|
|                 |                                 | The following events contain this property:<br><a href="#">slideLoadComplete</a> ,<br><a href="#">slideMetadataLoad</a> , <a href="#">slideChange</a>                                       |
| stepIndex       | Number                          | Animation step index.<br>The following events contain this property:<br><a href="#">stepChange</a>  |
| position        | Number                          | Slide playback position.<br>The following events contain this property:<br><a href="#">positionChange</a> , <a href="#">seekingComplete</a>   |
| succeeded       | Boolean                         | Indicates whether the bridge or player was loaded and initialized successfully.<br>The following events contain this property:<br><a href="#">bridgeLoaded</a> , <a href="#">playerInit</a> |
| player          | <a href="#">Player</a>          | The <a href="#">Player</a> class instance.<br>The following events contain this property:<br><a href="#">bridgeLoaded</a> , <a href="#">playerInit</a>                                      |
| volume          | Number                          | A sound volume value.<br>The following events contain this property:<br><a href="#">volumeChange</a> ,<br><a href="#">volumeChangingComplete</a>  |
| slideInfo       | <a href="#">SlideInfo</a>       | A <a href="#">SlideInfo</a> class instance.<br>The following events contain this property:<br><a href="#">slideMetadataLoad</a>   |
| soundController | <a href="#">SoundController</a> | A <a href="#">SoundController</a> class instance.<br>The following events contain this property:<br><a href="#">volumeChange</a> ,<br><a href="#">volumeChangingComplete</a>                |

**See also**

[PlaybackController Class](#)  
[Player Class](#)  
[SlideInfo Class](#)  
[SoundController Class](#)  
[Obsolete ActionScript 3 API](#)

**iSpring.as3bridge.BridgeLoader Class**

The **BridgeLoader** Class extends the standard [flash.events.EventDispatcher](#) class and exposes the following methods and events:

| Methods  |  |
|--|--|
| Method   | Description  |
| BridgeLoader(target: <a href="#">flash.display.DisplayObjectConta</a>  | Constructs a new BridgeLoader object which will load the bridge SWF module into the target DisplayObjectContainer object   |
| loadPresentation(bridgeURL:String, presentationURL:String, commandConnectionName:String = "", eventConnectionName:String = ""): <a href="#">Player</a> | Loads the ActionScript 3 bridge module and the iSpring generated Flash presentation.<br><b>Notes:</b> If command or event connection name is empty, it will be generated using current time stamp. |
| Events   |  |
| Type   | Description  |
| <a href="#">BridgeEvent.PLAYER_INIT</a>  | Occurs when player initialization is completed, just after the <a href="#">BridgeEvent.BRIDGE_LOADED</a> event.  |

**See also**

[BridgeEvent Class](#)  
[Player Class](#)  
[flash.events.EventDispatcher Class](#)  
[flash.display.DisplayObjectContainer Class](#)  
[Obsolete ActionScript 3 API](#)

**iSpring.as3bridge.PlaybackController Class**

The **PlaybackController** class extends the standard [flash.events.EventDispatcher](#) Class and provides the following properties, methods and events for presentation playback and navigation control.

| Properties                   |         |   |
|------------------------------|---------|---|
| Property                     | Type    | Description   |
| isPlaying                    | Boolean | Indicates whether the presentation is playing or not (read only)  |
| currentSlideDuration         | Number  | A duration of the currently playing slide (read only)   |
| currentSlideIndex            | Number  | Current slide index (read only)   |
| currentSlidePlaybackPosition | Number  | Returns playback position within current slide.<br><b>Note:</b> playback position is a number between 0 and 1 |
| currentStepIndex             | Number  | Returns current animation step index  |

| Methods  |   |
|--|---|
| Method   | Description   |
| play():Void  | Resumes presentation playback   |
| pause():Void                                       | Suspends presentation playback  |
| gotoNextSlide([autoStart:Boolean = true]):Void     | Switches to the next presentation slide and starts its playback.<br><b>Note:</b> if the autoStart parameter was set to false, slide playback would not be started after switching     |
| gotoPreviousSlide([autoStart:Boolean = true]):Void | Switches to the previous presentation slide and starts its playback.<br><b>Note:</b> if the autoStart parameter was set to false, slide playback would not be started after switching |

|  |   |
|--|---|
| <code>gotoLastViewedSlide([autoStart:Boolean = true]):Void</code>          | Switches to the previously viewed slide and starts its playback.<br><b>Note:</b> if the <b>autoStart</b> parameter was set to false, slide playback would not be started after switching  |
| <code>getCurrentSlideIndex():Number</code>                                 | Returns a zero-based index of currently playing slide   |
| <code>gotoSlide(slideIndex:Number, [autoStart:Boolean = true]):Void</code> | Switches to the given slide of the presentation and starts its playback<br><b>Notes:</b> Slide index is zero-based.<br>If the <b>autoStart</b> parameter was set to false, slide playback would not be started after switching      |
| <code>getCurrentSlidePlaybackPosition():Number</code>                      | Returns playback position within current slide.<br><b>Note:</b> playback position is a number between 0 and 1   |
| <code>pauseCurrentSlideAt(position:Number):Void</code>                     | Suspends currently playing slide at the given position<br><b>Note:</b> position is a number in between 0 and 1  |
| <code>playCurrentSlideFrom(position:Number):Void</code>                    | Restarts current slide from the given position.<br><b>Note:</b> position is a number between 0 and 1  |
| <code>seek(position:Number):Void</code>                                    | Seeks the current slide to the given position. After this operation slide playback gets suspended until the <b>endSeek()</b> method or other playback control method is called<br><b>Note:</b> position is a number between 0 and 1 |
| <code>endSeek([resumePlayback:Boolean = undefined]):Void</code>            | Exits from seeking mode and restores playback.<br><b>Note:</b> the <b>resumePlayback</b> parameter allows to start or pause slide after seeking explicitly  |
| <code>getCurrentStepIndex():Number</code>                                  | Returns current animation step index  |
| <code>gotoNextStep():Void</code>   | Plays current slide from the beginning of the next animation step.<br><b>Note:</b> being called at the end of the slide this method plays presentation from the beginning of the next slide   |
| <code>gotoPreviousStep():Void</code>                                       | Moves slide playback one step back. This method suspends slide playback at the end of the previous animation step.  |

|   |   |
|---|---|
|   | <b>Note:</b> being called on the start slide step this method causes playback switching to the end of the previous slide  |
| setAnimationStepPause(pause:Number):Void                    | This method allows setting a pause between animation steps. If an <b>undefined</b> value is passed as pause value, animation steps will not be played automatically |
| playFromStep(stepIndex:Number):Void                         | Plays current slide from the beginning of the specified animation step  |
| pauseAtStepStart(stepIndex:Number):Void                     | Suspends current slide at the beginning of the specified animation step   |
| pauseAtStepEnd(stepIndex:Number):Void                       | Suspends current slide at the end of specified animations step  |
| enableAutomaticSlideSwitching(enableSwitching:Boolean):Void | Enables or disables automatic switching to the next slide.<br><b>Note:</b> this method affects automatically switching slides only.                                 |
| getAutomaticSlideSwitching():Boolean                        | Returns true if automatic slide switching has been enabled, false otherwise.<br><b>Note:</b> automatic slide switching is enabled by default                        |

| Events                                      |   |
|---|---|
| Type  | Description   |
| <a href="#">BridgeEvent.PAUSE</a>           | Occurs when presentation playback is suspended                              |
| <a href="#">BridgeEvent.PLAY</a>            | Occurs when presentation playback is resumed                                |
| <a href="#">BridgeEvent.STEP_CHANGE</a>     | Occurs when currently playing animation step index is changed               |
| <a href="#">BridgeEvent.POSITION_CHANGE</a> | Occurs when the playback position of the currently playing slide is changed |

|   |   |
|---|---|
| <a href="#">BridgeEvent.SLIDE_CHANGE</a>        | Occurs when current slide is changed by another slide of the presentation |
| <a href="#">BridgeEvent.SLIDE_LOAD_COMPLETE</a> | Occurs when a slide of the presentation is loaded                         |
| <a href="#">BridgeEvent.PLAYBACK_COMPLETE</a>   | Occurs when the presentation playback is completed                        |
| <a href="#">BridgeEvent.SEEKING_COMPLETE</a>    | Occurs when a seek command execution is completed                         |

**See also**[IPlaybackListener Interface](#)[IPlayer Interface](#)[Flash Presentation ActionScript API Reference](#)**iSpring.as3bridge.Player Class**

The **Player** Class extends the standard [flash.events.EventDispatcher](#) class and exposes the following properties and events:

| Properties         |                                    |   |
|--------------------|------------------------------------|---|
| Property           | Type                               | Description   |
| playbackController | <a href="#">PlaybackController</a> | Returns the <a href="#">PlaybackController</a> class providing presentation playback control and navigation facilities  |
| soundController    | <a href="#">SoundController</a>    | Returns the <a href="#">SoundController</a> class providing sound volume control facilities   |
| presentationInfo   | <a href="#">PresentationInfo</a>   | Returns the <a href="#">PresentationInfo</a> class providing the presentation content information   |
| initialized        | Boolean                            | Returns a Boolean value indicating whether the player was initialized. The following properties return an <b>undefined</b> value until the player gets initialized: |

|  |  |   |
|--|--|---|
|  |  | <ul style="list-style-type: none"> <li>• presentationInfo</li> <li>• playbackController</li> <li>• soundController</li> </ul> |
|--|--|---|

| Events                                    |   |
|---|---|
| Type                                      | Description   |
| <a href="#">BridgeEvent.BRIDGE_LOADED</a> | Occurs when the ActionScript bridge module is loaded and ready to receive commands from the parent Flash movie. |

**See also**

- [PlaybackController Class](#)
- [PresentationInfo Class](#)
- [SoundController Class](#)
- [flash.events.EventDispatcher Class](#)
- [Obsolete ActionScript 3 API](#)

**iSpring.flex.PresentationContainer Class**

The **PresentationContainer** class extends the standard [mx.core.UIComponent](#) class and provides the following properties:

| Properties   |                              |  |
|--------------|------------------------------|--|
| Property     | Type                         | Description  |
| bridgeLoader | <a href="#">BridgeLoader</a> | Returns a <a href="#">BridgeLoader</a> object providing an ability to load iSpring generated presentations |

**Samples**

This class can be used in Flex applications to put a presentation container UI component to a Flex form.

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute"
creationComplete="{creationComplete()}"
xmlns:mc="ispring.flex.*" frameRate="30" minWidth="470"
minHeight="200">
  <mx:Script source="main.as"/>
  <mc:PresentationContainer id="presentationContainer" width="780"
```

```
height="524" x="10" y="36">
  </mc:PresentationContainer>
  <!-- the rest UI elements are placed here-->
</mx:Application>
```

The following code loads the generated presentation into the presentation container (file main.as).

```
import iSpring.flex.PresentationContainer;
import iSpring.as3bridge.*;

private function creationComplete():void
{
    var bridgeLoader:BridgeLoader = presentationContainer.bridgeLoader;
    bridgeLoader.loadPresentation("as3bridge.swf", "presentation.swf");
}
```

### See also

[BridgeLoader Class](#)

[mx.core.UIComponent Class](#)

[Obsolete ActionScript 3 API](#)

#### iSpring.as3bridge.PresentationInfo Class

The **PresentationInfo** class stores the presentation information and provides the following properties:

| Properties    |                                  |  |
|---------------|----------------------------------|--|
| Property      | Type                             | Description  |
| hasPresenter  | Boolean                          | Indicates whether the presentation contains Presenter information                                      |
| presenterInfo | <a href="#">PresenterInfo</a>    | Returns the <a href="#">PresenterInfo</a> Class providing the information about presentation presenter |
| slides        | <a href="#">SlidesCollection</a> | Returns the <a href="#">SlidesCollection</a> Class providing information about presentation slides     |
| hasReferences | Boolean                          | Indicates whether the presentation contains references   |

|             |                                      |  |
|-------------|--------------------------------------|--|
| references  | <a href="#">ReferencesCollection</a> | Returns the <a href="#">ReferencesCollection</a> Class providing the information about presentation references   |
| title       | String                               | Presentation title   |
| duration    | Number                               | Presentation duration in seconds   |
| slideWidth  | Number                               | Presentation slide width in pixels   |
| slideHeight | Number                               | Presentation slide height in pixels  |
| frameRate   | Number                               | Presentation frame rate (number of frames per second)<br><b>Note:</b> when iSpring generated presentation is loaded by another Flash movie, it always plays at its frame rate. |

**See also**[PresenterInfo Class](#)[SlidesCollection Class](#)[ReferencesCollection Class](#)[Obsolete ActionScript 3 API](#)**iSpring.as3bridge.PresenterInfo Class**

The **PresenterInfo** Class stores the information about presenter and provides the following properties:

| Properties    |        |                                     |
|---------------|--------|-------------------------------------|
| Property      | Type   | Description                         |
| name          | String | Presenter name                      |
| title         | String | Presenter title                     |
| biographyText | String | A brief information about presenter |

|         |        |                            |
|---------|--------|----------------------------|
| email   | String | Presenter e-mail address   |
| webSite | String | Presenter Web site address |

**See also**

[PresentationInfo Class](#)

[Obsolete ActionScript 3 API](#)

**iSpring.as3bridge.ReferenceInfo Class**

The **ReferenceInfo** Class provides the following properties for retrieving reference information:

| Properties |        |   |
|------------|--------|---|
| Property   | Type   | Description   |
| title      | String | Reference title   |
| url        | String | Reference URL   |
| target     | String | Target window name where the reference should be opened |

**See also**

[ReferencesCollection Class](#)

[Obsolete ActionScript 3 API](#)

**iSpring.as3bridge.ReferencesCollection Class**

The **ReferencesCollection** class provides the following properties and methods for accessing presentation references:

| Properties      |        |  |
|-----------------|--------|--|
| Property        | Type   | Description                            |
| referencesCount | Number | Number of references in the collection |

| Methods   |  |
|---|--|
| Method  | Description  |
| getReference(index:Number): <a href="#">ReferenceInfo</a> | Returns <a href="#">ReferenceInfo</a> class providing the information about the specified reference.<br><b>Note:</b> the <b>index</b> parameter is an integer value from 0 to (number of references minus 1) |

**See also**

[ReferenceInfo Class](#)  
[PresentationInfo Class](#)  
[Obsolete ActionScript 3 API](#)

**iSpring.as3bridge.SlideInfo Class**

The **SlideInfo** Class provides information about the particular slide of the presentation such as title, duration, text, etc. It exposes the following properties and methods.

| Properties     |                                |   |
|----------------|--------------------------------|---|
| Property       | Type                           | Description   |
| animationSteps | <a href="#">AnimationSteps</a> | Returns the <a href="#">AnimationSteps</a> class providing the information about slide animation steps  |
| metadataLoaded | Boolean                        | Indicates whether slide metadata has been loaded<br><b>Note:</b> all slide properties except <b>startTime</b> property are undefined until slide metadata is loaded |
| isLoading      | Boolean                        | Indicates whether slide has been loaded<br><b>Note:</b> it is guaranteed that loaded slide metadata is loaded   |
| title          | String                         | Slide title   |
| duration       | Number                         | Slide duration in seconds   |

|                     |        |  |
|---------------------|--------|--|
| notesText           | String | Slide notes text   |
| startTime           | Number | Slide start time within presentation timeline  |
| endTime             | Number | Slide end time within presentation timeline  |
| startStepIndex      | Number | Slide start step index within the presentation   |
| endStepIndex        | Number | Slide end step index within the presentation   |
| slideText           | String | Slide text in the normalized form (see the <a href="#">remarks</a> section)              |
| notesTextNormalized | String | Slide notes text in the <b>normalized form</b> (see the <a href="#">remarks</a> section) |
| titleNormalized     | String | Slide title in the <b>normalized form</b> (see the <a href="#">remarks</a> section)      |

Normalized strings are strings having all characters in the lower case with no extra spaces between words. This simplifies and speeds up full text search implementation.

### See also

[AnimationSteps Class](#)

[SlidesCollection Class](#)

[Obsolete ActionScript 3 API](#)

#### **iSpring.as3bridge.SlidesCollection Class**

The **ISlidesCollection** interface represents a collection of presentation slides. It provides the following properties and methods:

| Properties |      |             |
|------------|------|-------------|
| Property   | Type | Description |
|            |      |             |

|             |        |                                      |
|-------------|--------|--------------------------------------|
| slidesCount | Number | Number of slides in the presentation |
|-------------|--------|--------------------------------------|

| Methods  |   |
|--|---|
| Method   | Description   |
| getSlideInfo(slideIndex:Number): <a href="#">SlideInfo</a> | Returns <a href="#">SlideInfo</a> Class providing information about the specified slide<br><b>Note:</b> Slide index is zero-based |

**See also**[PresentationInfo Class](#)[SlideInfo Class](#)[Obsolete ActionScript 3 API](#)**iSpring.as3bridge.SoundController Class**

The **SoundController** class provides sound volume control facilities. It extends the standard [flash.events.EventDispatcher](#) class and provides the following properties and events.

| Properties |        |   |
|------------|--------|---|
| Property   | Type   | Description   |
| volume     | Number | Sound volume<br><b>Note:</b> sound volume is a number between 0 and 1 |

| Events   |  |
|--|--|
| Type   | Description  |
| <a href="#">BridgeEvent.VOLUME_CHANGE</a>            | Occurs when the sound volume is changed  |
| <a href="#">BridgeEvent.VOLUME_CHANGING_COMPLETE</a> | Occurs when the sound volume is changed via <b>volume</b> property of the SoundController object |

**See also**[Player Class](#)

[flash.events.EventDispatcher Class](#)  
[Obsolete ActionScript 3 API](#)

### 1.5.3. Skin Development Reference

#### 1.5.3.1. fsplayer.ui.skins.ISkin Interface

The **ISkin** interface is provided by all iSpring-compatible skin modules and used to connect the player core with the User Interface. The **ISkin** interface exposes the following methods called by the player core.

| Methods  |   |
|--|---|
| Method   | Description   |
| initialize(player: <a href="#">IPlayer</a> ):Void          | This method is invoked by the player core to initialize the skin. The information provided by the passed <b>player</b> parameter can be used by the skin to initialize its visual components.<br><b>Note:</b> playback control methods of the <a href="#">IPresentationPlaybackController</a> interface are inaccessible until the skin gets initialized. Skins use <a href="#">ISkinListener</a> interface to notify the core at the end of the skin initialization process. See <a href="#">Remarks</a> section for more details. |
| getSlideShowWindow(): <a href="#">ISlideShowWindow</a>     | Returns the <a href="#">ISlideShowWindow</a> interface providing the player core with the target MovieClip for presentation slides loading.   |
| setListener(listener: <a href="#">ISkinListener</a> ):Void | The player core passes <a href="#">ISkinListener</a> interface that allows core to receive notifications about skin initialization process.   |

#### Remarks

Skin initialization can take some time, especially when the skin loads some extra data from the Internet. In order to get the initialization notification the player core implements the [ISkinListener](#) Interface and passes it to the skin via the **setListener()** method.

#### Samples

The following example illustrates the minimal implementation of the **ISkin**,

[ISlideShowWindow](#), an [IPlaybackListener](#) interfaces.

```
import fsplayer.ui.skins.*;
import fsplayer.api.*;

class CSkinImpl implements IPlaybackListener, ISkin, ISlideShowWindow
{
private var m_skinTarget:MovieClip;

private var m_slideAssetsTargetMovieClip:MovieClip;

private var m_listener:ISkinListener;
private var m_player:IPlayer;
private var m_skinSettings:Object;
private var m_presentation:IPresentationInfo;
private var m_playbackController:IPresentationPlaybackController;
private var m_skinInitialized:Boolean;

public function CSkinImpl(target:MovieClip)
{
    m_skinTarget = target;
}

// This method is called to initialize the skin
function initialize(playerAPI:IPlayer):Void
{
    m_player = playerAPI;
    m_skinSettings = m_player.getSettings();
    m_playbackController = m_player.getPlaybackController();
    m_presentation = m_player.getPresentationInfo();

    m_playbackController.addListener(this);
}

// returns slide show window
function getSlideShowWindow():ISlideShowWindow
{
    return this;
}

// ISlideShowWindow.getTarget() returns the target MovieClip where slides
// will be loaded and displayed by player core
function getTarget():MovieClip
{
    if (!m_slideAssetsTargetMovieClip)
    {
        m_slideAssetsTargetMovieClip = createSlideAssetsTarget();
    }

    return m_slideAssetsTargetMovieClip;
}

// this method is called by playback core to get notified when the skin is
```

```
loaded
function setListener(listener:ISkinListener):Void
{
    m_listener = listener;
    if (!m_skinInitialized)
    {
        m_skinInitialized = true;
        m_listener.onSkinInit(this);
    }
}

// empty implementation of IPlaybackListener interface methods. They can be
// overridden by inherited classes
function onPausePlayback():Void{}
function onStartPlayback():Void{}
function onCurrentSlideIndexChanged(slideIndex:Number):Void{}
function onSlideLoadingComplete(slideIndex:Number):Void{}
function onSlidePositionChanged(position:Number):Void{}
function onAnimationStepChanged(stepIndex:Number):Void{}
function onPresentationPlaybackComplete():Void{}

// This method can be used in the inherited classes to get the skin
// settings
private function getSkinSettings():Object
{
    return m_skinSettings;
}

// returns the target MovieClip where the skin resides. This method can be
// overridden in inherited classes.
private function getSkinTarget():MovieClip
{
    return m_skinTarget;
}

// This method creates the MovieClip slides will be loaded into
// This method can be overridden in the inherited classes
private function createSlideAssetsTarget():MovieClip
{
    return getSkinTarget().createEmptyMovieClip("assets",
getSkinTarget().getNextHighestDepth());
}
}
```

**See also**

[IPlayer Interface](#)

[IPresentationPlaybackController Interface](#)

[ISkinListener Interface](#)

[ISlideShowWindow Interface](#)

[Skin development](#)

### 1.5.3.2. fsplayer.ui.skins.ISkinListener Interface

The **ISkinListener** interface is implemented by the player core and allows it to receive the notification from the skin when the skin has been initialized.

| Methods                                       |  |
|---|--|
| Method  | Description  |
| onSkinInit(skin: <a href="#">ISkin</a> ):Void | This method is invoked by skins to notify the player core that the skin has been initialized. The <b>skin</b> parameter is an object which implements the <a href="#">ISkin</a> interface. |

#### Remarks

Skin initialization can require some time especially if the skin needs to load some information from the Internet in order to finish the initialization process. Skin notifies the player core via **onSkinInit()** method call on the **ISkinListener** interface. The **ISkinListener** Interface is passed to the skin via **setListener()** method of the [ISkin](#) Interface.

The player core will not start loading presentation slides until it gets **onSkinInit** notification.

#### See also

[ISkin Interface](#)  
[Skin development](#)

### 1.5.3.3. fsplayer.ui.skins.ISkinSizeController Interface

The **ISkinSizeController** interface provides methods for resizing, scaling and self-scaling of player skin. Advanced, Crystal and Lite skins currently implement support of this interface.

| Methods                            |  |
|------------------------------------|--|
| Method                             | Description  |
| getAutoSize():Boolean              | Returns true if self-scaling is been enabled for the skin, false otherwise   |
| setAutoSize(autoSize:Boolean):Void | This method allows to set self-scaling property. If self-scaling is enabled, the skin will occupy entire Flash player window |

|   |  |
|---|--|
| getWidth():Number                           | Returns width of the skin in pixels  |
| getHeight():Number                          | Returns height of the skin in pixels   |
| setSize(width:Number, height:Number):Void   | This method allows to set width and height in pixels of the window where skin is displayed. Applies to the skin if self-scaling is disabled                        |
| getScaleX():Number                          | Returns X-direction scaling value of the window where skin is displayed. Applies to the skin if self-scaling is disabled<br><b>Note:</b> The default value is 100% |
| getScaleY():Number                          | Returns Y-direction scaling value of the window where skin is displayed. Applies to the skin if self-scaling is disabled<br><b>Note:</b> The default value is 100% |
| setScale(scaleX:Number, scaleY:Number):Void | This method allows to set X and Y scaling of the window where skin is displayed  |

**See also**[ISkin Interface](#)**1.5.3.4. fsplayer.ui.skins.ISlideShowWindow Interface**

The **ISlideShowWindow** interface provides the player core with the target MovieClip for presentation slides loading.

| Methods               |  |
|-----------------------|--|
| Method                | Description  |
| getTarget():MovieClip | Returns the MovieClip which will be used by the player core to load and display slides |

**Remarks**

iSpring generated presentations can be wrapped into one of the existing skins or into a custom made skin. Each skin provides its unique user interface. Slide loading and playback is a responsibility of the player core while UI and navigation is a responsibility of skins. The **ISlideShowWindow** interface provides playback core with the necessary target MovieClip

for presentation slides loading and display.

**See also**

[ISkin Interface](#)  
[Skin development](#)

### 1.5.4. Flash Slide ActionScript API Reference

#### 1.5.4.1. slideplayer.api.IHyperlinksManager Interface

The **IHyperlinksManager** Interface provides slides with an ability to switch to another slides or open external web pages. A standalone slide does not switch to another slide when a hyperlink is clicked. An external Flash application can use `setHyperlinksManager()` method of the [ISlideController](#) interface to pass an object implementing the **IHyperlinksManager** interface which receive notifications when user clicks a hyperlink or action button.

The following methods are exposed by the **IHyperlinksManager** interface.

| Methods  |   |
|--|---|
| Method   | Description   |
| <code>gotoSlide(slideIndex:Number):Void</code>       | This method is invoked when user clicks a hyperlink to another slide<br>An implementation can hide the current slide and load and show the new one  |
| <code>openURL(url:String, target:String):Void</code> | This method is invoked when user clicks a hyperlink to an external web page or other resource<br><b>Note:</b> the simplest way to implement this method is to call following ActionScript function:<br><code>getURL(url, target)</code> |
| <code>gotoLastViewedSlide():Void</code>              | This method is invoked when user clicks a hyperlink to the last viewed slide.<br>An implementation must hide the current slide and show the slide that was shown before   |
| <code>endShow():Void</code>                          | This method is invoked when user clicks a hyperlink that finishes slide show<br>Typically, an implementation unloads all slides of the current presentation   |

## Samples

The following example illustrates a typical IHyperlinksManager interface implementation and shows its usage.

### CMyHyperlinksManager.as

```
import slideplayer.api.*;

class CMyHyperlinksManager implements IHyperlinksManager
{
    private var m_viewedSlides:Array;
    private var m_currentSlideIndex:Number;

    function CMyHyperlinksManager()
    {
        m_viewedSlides = new Array();
    }

    function gotoSlide(slideIndex:Number):Void
    {
        m_currentSlideIndex = slideIndex;
        m_viewedSlides.push(slideIndex);
        loadSlide(slideIndex);
    }

    function loadSlide(slideIndex:Number):Void
    {
        // empty
    }

    function openURL(url:String, target:String):Void
    {
        getURL(url, "_blank");
    }

    function gotoLastViewedSlide():Void
    {
        if (m_viewedSlides.length > 1)
        {
            var curSlide:Number = Number(m_viewedSlides.pop());
            var lastViewedSlide:Number;

            do
            {
                lastViewedSlide = Number(m_viewedSlides.pop());
            } while ((lastViewedSlide == curSlide) &&
(m_viewedSlides.length != 0));

            if (lastViewedSlide != curSlide)
            {
                gotoSlide(lastViewedSlide);
            }
        }
    }
}
```

```
        }
        else
        {
            m_viewedSlides.push(curSlide);
        }
    }
}

function endShow():Void
{
    fscommand("quit", "true");
}
}
```

CMyHyperlinksManager class usage example:

```
import slideplayer.api.*;
System.security.allowDomain("*");

var rootMC:MovieClip = this;
var slidesMC:MovieClip = rootMC.createEmptyMovieClip("slidesMC", 2);
var slideMC1:MovieClip = slidesMC.createEmptyMovieClip("slide1", 1);

var g_slideController:ISlideController;
var g_hyperlinksManager:CMyHyperlinksManager = new CMyHyperlinksManager();

g_hyperlinksManager.loadSlide = function(slideIndex:Number):Void
{
    loadSlide(slideIndex);
}

g_hyperlinksManager.gotoSlide(0);

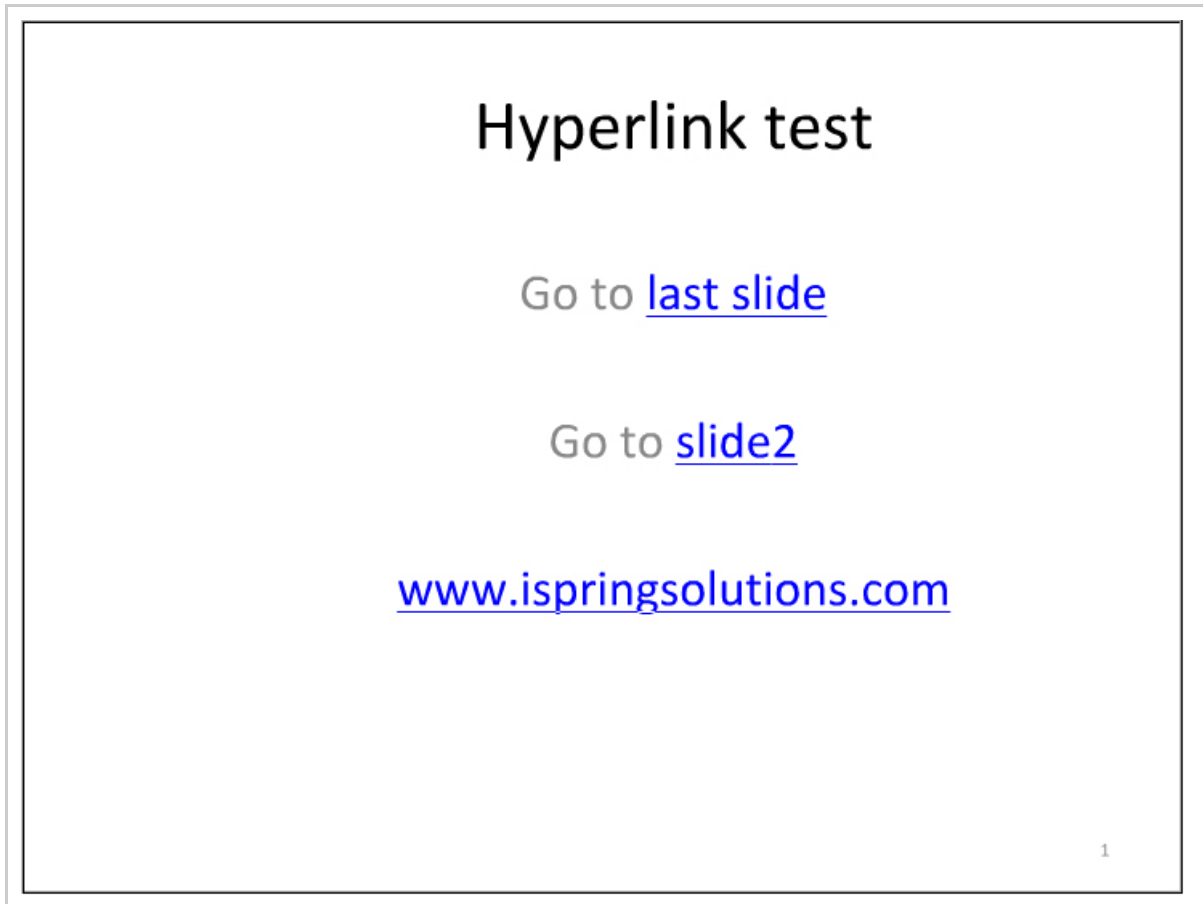
// loads slide with the specified slide index
function loadSlide(slideIndex:Number):Void
{
    // remove hyperlinks manager from the current slide
    g_slideController.setHyperlinksManager(undefined);

    var ml:MovieClipLoader = new MovieClipLoader();
    ml.addListener(this);
    ml.loadClip("data_hl/slide" + (slideIndex + 1) + ".swf", slideMC1);
}

function onLoadInit(target:MovieClip):Void
{
    g_slideController = target.getSlideController();
    slideController.setHyperlinksManager(g_hyperlinksManager);
}
```

## Demo

The following demo illustrates hyperlinks and action buttons handling via **IHyperlinksManager** interface.



**See also**

[ISlideController Interface](#)

[Flash Slide ActionScript API Reference](#)

#### **1.5.4.2. slideplayer.api.IPlaybackController Interface**

The **IPlaybackController** interface provides methods for slide playback control. This interface can be obtained by calling **getPlaybackController()** method on [ISlideController](#) interface.

The IPlaybackController interface exposes the following methods:

| Methods   |   |
|---|---|
| Method  | Description   |
| addListener(listener:IPlaybackListener):Void    | Adds an external object implementing <a href="#">IPlaybackListener</a> interface; it will be notified about events during playback; the passed object will receive slide playback event notifications |
| removeListener(listener:IPlaybackListener):Void | Removes previously added listener object. The passed object won't receive notifications from the playback controller any more   |
| isPlaying():Boolean                             | Returns a Boolean value indicating whether the slide is playing or not  |
| play():Void                                     | Resumes slide playback  |
| pause():Void                                    | Suspends slide playback   |
| getPosition():Number                            | Returns slide playback position<br><b>Note:</b> playback position is a number between 0 and 1   |
| playFromPosition(position:Number):Void          | Plays the slide from specified position.<br><b>Note:</b> playback position is a number between 0 and 1  |
| pauseAtPosition(position:Number):Void           | Suspends slide at the specified position.<br><b>Note:</b> playback position is a number between 0 and 1   |
| getCurrentStepIndex():Number                    | Returns current animation step index  |
| gotoNextStep():Void                             | Plays the slide from the beginning of the next animation step   |
| gotoPreviousStep():Void                         | Moves slide playback one step back. This method suspends slide playback at the end of the previous animation step   |
| playFromStep(stepIndex:Number):Void             | Plays the slide from the beginning of the specified animation step  |

|   |   |
|---|---|
| pauseAtStepStart(stepIndex:Number):Void | Suspends the slide at the beginning of the specified animation step |
| pauseAtStepEnd(stepIndex:Number):Void   | Suspends the slide at the end of specified animations step          |

## Samples

The following example illustrates how to obtain and use the `IPlaybackController` interface to control slide playback

```
import slideplayer.api.*;

// slide loading and initialization code is placed here
// ...

var g_slideController:ISlideController;
var g_playbackController:IPlaybackController;

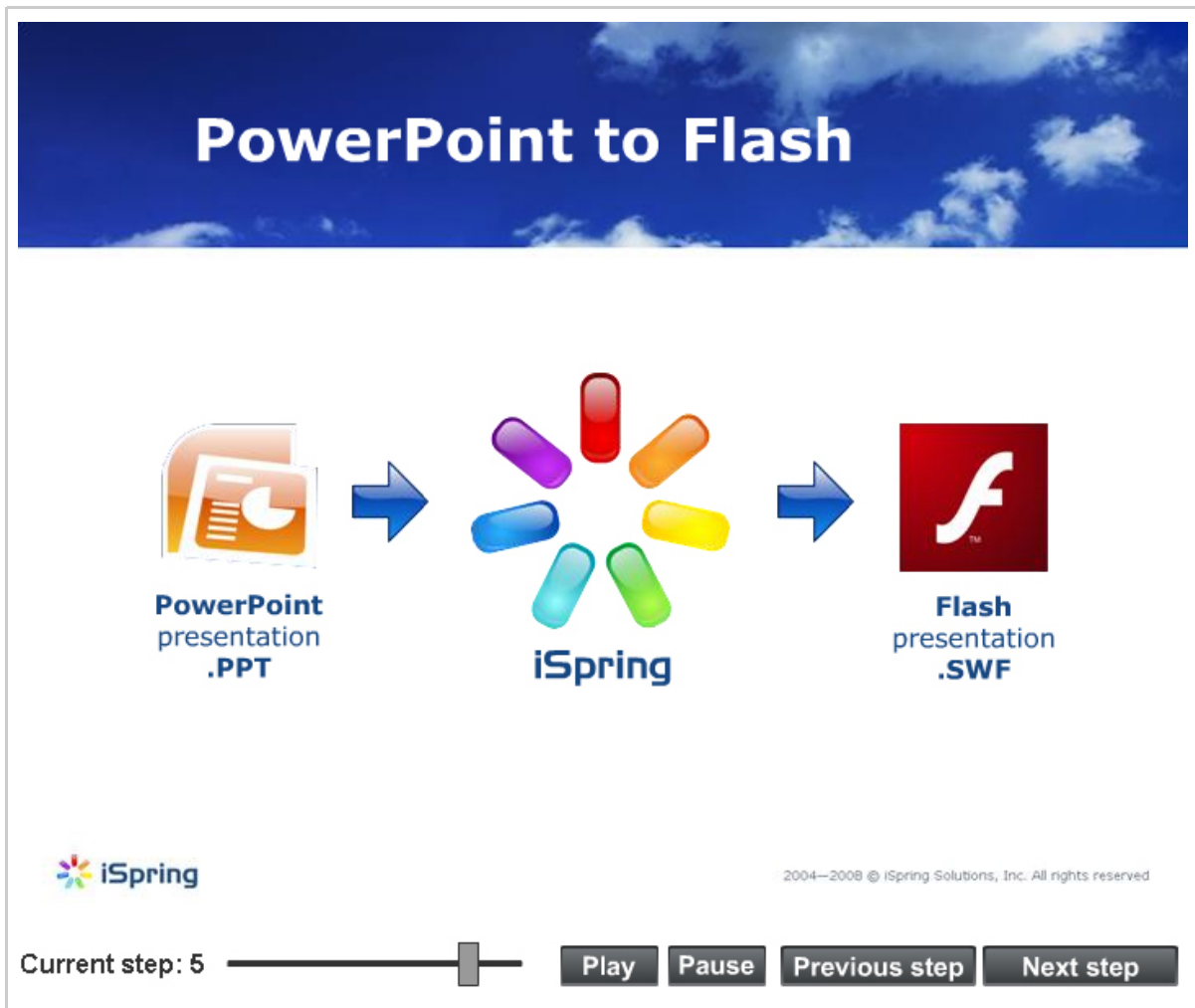
// this function is called when slide movie clip has been loaded
function onLoadInit(target:MovieClip):Void
{
    g_slideController = target.getSlideController();
    g_playbackController = g_slideController.getPlaybackController();
}

playPauseSlideButton.onRelease = function()
{
    if (g_playbackController)
    {
        if (g_playbackController.isPlaying())
        {
            g_playbackController.pause();
        }
        else
        {
            g_playbackController.play();
        }
    }
}
```

## Demo

This demo illustrates slide playback control via **IPlaybackController** interface.

---



## References

[IPlaybackListener Interface](#)

[ISlideController Interface](#)

[Flash Slide ActionScript API Reference](#)

### 1.5.4.3. slideplayer.api.IPlaybackListener Interface

There are various events occurring during slide playback. The playback controller notifies external objects about these events via **IPlaybackListener** interface. This interface defines a set of methods which are invoked by playback controller when the corresponding events occur. These methods are as follows.

| Methods                                       |  |
|---|--|
| Method  | Description  |
| onPausePlayback():Void                        | Invoked when presentation playback is suspended              |
| onStartPlayback():Void                        | Invoked when presentation playback is resumed                |
| onSlidePositionChanged(position:Number):Void  | Invoked when playback position is changed during playback    |
| onAnimationStepChanged(stepIndex:Number):Void | Invoked when slide animation step is changed during playback |
| onSlidePlaybackFinished():Void                | This method is called when slide playback was finished.      |

In order to receive event notifications you need to create a class which implements **IPlaybackListener** interface and pass the instance of this class to the **addListener** method of the [IPlaybackController](#) interface.

#### See also

[IPlaybackController Interface](#)  
[Flash Slide ActionScript API Reference](#)

#### 1.5.4.4. slideplayer.api.ISlideController Interface

The **ISlideController** interface is exposed by each standalone slide. It provides the following methods.

| Methods  |   |
|--|---|
| Method   | Description   |
| getPlaybackController(): <a href="#">IPlaybackController</a> | Returns the <a href="#">IPlaybackController</a> interface providing slide playback control facilities |
| getSoundController(): <a href="#">ISoundController</a>       | Returns the <a href="#">ISoundController</a> interface providing sound volume control facilities      |
| getHyperlinkManager(): <a href="#">IHyperlinksManager</a>    | Returns the <a href="#">IHyperlinksManager</a> interface providing                                    |

|   |   |
|---|---|
|   | hyperlink action handling facilities  |
| setHyperlinksManager(manager: <a href="#">IHyperlinksManager</a> ): | Sets the new <a href="#">IHyperlinksManager</a> interface which will handle hyperlink actions |
| getSlideWidth():Number  | Returns slide width in pixels   |
| getSlideHeight():Number   | Returns slide height in pixels  |
| getStepsCount():Number  | Returns number of animation steps on the slide  |

### Samples

The following example illustrates how to load a flash slide and get its **ISlideController** interface.

```
import slideplayer.api.*;
System.security.allowDomain("*");

// create a movie clip where slide will be loaded
var slideMC1:MovieClip = this.createEmptyMovieClip("slide1", 1);

// load slide with name slide1.swf into slideMC1 movie clip
var ml:MovieClipLoader = new MovieClipLoader();
ml.addListener(this);
ml.loadClip("slide1.swf", slideMC1);

var g_slideController:ISlideController;

// this function is invoked by MovieClipLoader when the loaded Flash slide
is initialized
function onLoadInit(target:MovieClip):Void
{
    // now we can get ISlideController interface and access its methods
    g_slideController = target.getSlideController();
}
```

### Demo

The following demo illustrates standalone slide loading and playback control via [IPlaybackController](#) interface.





The image shows a 3D rendering of the iSpring Pro 3.1 software box on the left. The box is blue with a colorful logo of ten dots in a circle. Text on the box includes 'iSpring Pro 3.1', 'PowerPoint to Flash converter', and 'iSpring Pro 3.1'. To the right of the box is a banner with a blue sky and white clouds background. The banner text reads 'iSpring Pro 3.1', 'PowerPoint to Flash', and 'Easy. Fast. Accurate.'. At the bottom of the banner, it says '2004—2008 © iSpring Solutions, Inc. All rights reserved.' Below the banner is a control bar with a 'Play/Pause' button, a 'Load slide' label, and nine numbered buttons (1-9).

**See also**

- [IHyperlinksManager Interface](#)
- [IPlaybackController Interface](#)
- [ISoundController Interface](#)
- [Flash Slide ActionScript API Reference](#)

**1.5.4.5. slideplayer.api.ISoundController Interface**

The **ISoundController** Interface provides the following methods for sound control.

|         |
|---------|
| Methods |
|---------|

| Method                        | Description  |
|-------------------------------|--|
| getVolume():Number            | Returns current sound volume.<br><b>Note:</b> sound volume is a number between 0 and 1 |
| setVolume(volume:Number):Void | Sets sound volume.<br><b>Note:</b> sound volume is a number between 0 and 1            |

### Samples

```
var g_slideController:ISlideController;
var g_soundController:ISoundController;

function loadSlide(slideIndex:Number):Void
{
    var ml:MovieClipLoader = new MovieClipLoader();
    ml.addListener(this);
    var fileName:String = "data_snd/slide" + slideIndex + ".swf";

    var slashPos:Number = Math.max(this._url.lastIndexOf("/"),
this._url.lastIndexOf("\\\\"));
    var fullPath:String = this._url.substr(0, slashPos + 1) + fileName;
    ml.loadClip(fullPath, slideMC1);
    loadingLabel.text = this._url;
}

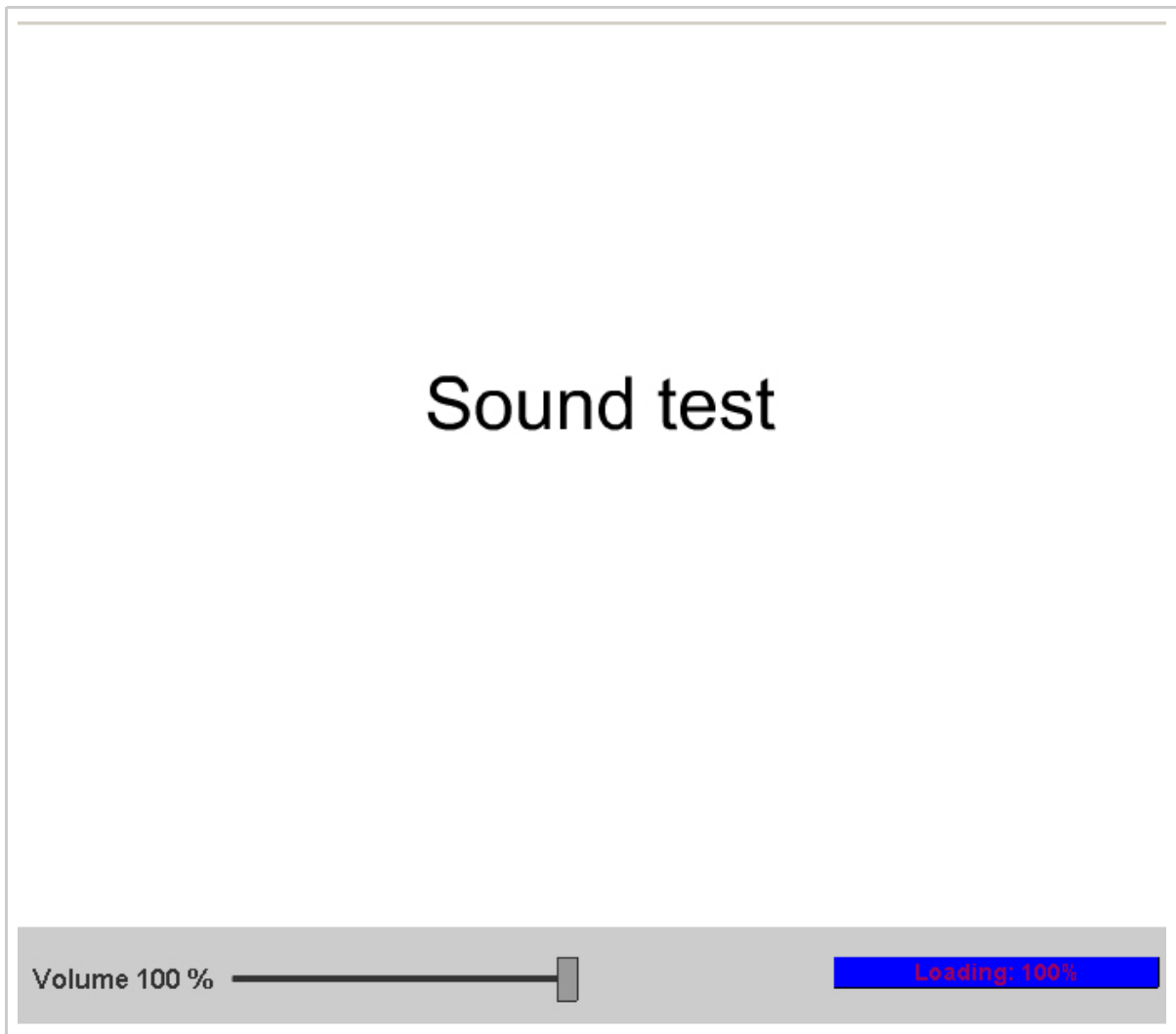
function onLoadInit(target:MovieClip):Void
{
    g_slideController = target.getSlideController();
    g_soundController = g_slideController.getSoundController();
    g_trackBar.setPos(g_soundController.getVolume());
}

g_trackBar.newPos = function(pos:Number):Void
{
    g_soundController.setVolume(pos);
}
```

### Demo

The following demo illustrates ISoundController interface usage to control sound volume on the slide.





**See also**

[ISlideController Interface](#)

[Flash Slide ActionScript API Reference](#)

## **1.6. JavaScript API Reference**

### **1.6.1. fsplayer.api.Player Class**

The **Player** class exposes the following methods:

| Methods                 |   |
|-------------------------|---|
| Method                  | Description   |
| getPlaybackController() | Returns the <a href="#">PlaybackController</a> object providing #he presentation content information. |
| getPresentationInfo()   | Returns the <a href="#">PresentationInfo</a> object providing #he presentation content information.   |

**See also**

- [PlaybackController Class](#)
- [PresentationInfo Class](#)
- [JavaScript support in iSpring](#)

**1.6.2. iSpring.as2player.PresentationLoader Class**

The **PresentationLoader** class provide facilities for loading iSpring generated Flash presentation and exposes the following methods:

| Methods  |   |
|--|---|
| Method   | Description   |
| load(movieName, contentParent, movieId, width, height) | Creates SWFObject with name specified by <b>movieName</b> and loads the iSpring generated Flash presentation in it. Flash object is placed in the content element with id specified by <b>contentParent</b> . <b>Width, height</b> parameters define dimensions and <b>movieId</b> defines Id of created SWFObject. |
| onPlayerInit(player: Object)                           | This method is calls after player was initialized. It can be overridden.  |
| getMovie()   | Returns SWFObject which contains iSpring generated Flash presentation.  |
| getMovieId()   | Returns identifier of SWFObject element which contains iSpring generated Flash presentation.  |

**See also**

[JavaScript support in iSpring](#)

**1.6.3. fsplayer.api.PlaybackController Class**

The **PlaybackController** class exposes the following methods:

| Constants           |        |  |   |
|---------------------|--------|--|---|
| Constant            | Type   | Value                                    | Description   |
| PAUSE_PLAYBACK      | String | playbackListener_onPause                 | Defines the name of pause playback event.                     |
| START_PLAYBACK      | String | playbackListener_onStartP                | Defines the name of start playback event.                     |
| ANIMATION_STEP_CHA  | String | playbackListener_onAnim                  | Defines the name of animation step changed event.             |
| SLIDE_POSITION_CHAN | String | playbackListener_onSlideF                | Defines the name of slide position changed event.             |
| CURRENT_SLIDE_INDE  | String | playbackListener_onCurre                 | Defines the name of current slide is changed event.           |
| SLIDE_LOADING_COM   | String | playbackListener_onSlideL                | Defines the name of slide loading is completed event.         |
| PRESENTATION_PLAYI  | String | playbackListener_onPreser                | Defines the name of presentation playback is completed event. |
| Methods             |        |  |   |
| Method              |        | Description                              |   |
| play()              |        | Starts or resumes presentation playback. |   |

|                                   |  |
|-----------------------------------|--|
| pause()                           | Suspends presentation playback.  |
| gotoSlide(slideIndex, autoStart)  | Switches to the given slide of the presentation and starts its playback<br><b>Notes:</b> Slide index is zero-based.  |
| isPlaying(slideIndex, autoStart)  | Indicates whether the presentation is playing or not.  |
| gotoNextSlide(autoStart)          | Switches to the next presentation slide and starts its playback. <b>Note:</b> if the autoStart parameter was set to false, slide playback would not be started after switching.  |
| gotoPreviousSlide(autoStart)      | Switches to the previous presentation slide and starts its playback. <b>Note:</b> if the autoStart parameter was set to false, slide playback would not be started after switching.  |
| getCurrentSlideIndex()            | Returns current slide index.   |
| getCurrentSlidePlaybackPosition() | Returns playback position within current slide. <b>Note:</b> playback position is a number between 0 and 1.  |
| gotoNextStep()                    | Plays current slide from the beginning of the next animation step. <b>Note:</b> being called at the end of the slide this method plays presentation from the beginning of the next slide.  |
| gotoPreviousStep()                | Moves slide playback one step back. This method suspends slide playback at the end of the previous animation step. <b>Note:</b> being called on the start slide step this method causes playback switching to the end of the previous slide. |
| getCurrentStepIndex()             | Returns current animation step index.  |
| setPausePlaybackHandler()         | Set <b>onPausePlayback()</b> method as handler for PAUSE_PLAYBACK event.   |
| onPausePlayback()                 | Does nothing. This method can be overridden.   |
| setStartPlaybackHandler()         | Set <b>onStartPlayback()</b> method as handler for   |

|  |   |
|--|---|
|  | START_PLAYBACK event.   |
| onStartPlayback()                        | Does nothing. This method can be overridden.  |
| setAnimationStepChangedHandler()         | Set <b>onAnimationStepChanged()</b> method as handler for ANIMATION_STEP_CHANGED event.   |
| onAnimationStepChanged(stepIndex)        | Does nothing. This method can be overridden.  |
| setSlidePositionChangedHandler()         | Set <b>onSlidePositionChanged()</b> method as handler for SLIDE_POSITION_CHANGED event.   |
| onSlidePositionChanged(position)         | Does nothing. This method can be overridden.  |
| setCurrentSlideIndexChangedHandler()     | Set <b>onCurrentSlideIndexChanged()</b> method as handler for CURRENT_SLIDE_INDEX_CHANGED event.  |
| onCurrentSlideIndexChanged(slideIndex)   | Does nothing. This method can be overridden.  |
| setSlideLoadingCompleteHandler()         | Set <b>onSlideLoadingComplete()</b> method as handler for SLIDE_LOADING_COMPLETE event.   |
| onSlideLoadingComplete(slideIndex)       | Does nothing. This method can be overridden.  |
| setPresentationPlaybackCompleteHandler() | Set <b>onPresentationPlaybackComplete()</b> method as handler for PRESENTATION_PLAYBACK_COMPLETE event.   |
| onPresentationPlaybackComplete()         | Does nothing. This method can be overridden.  |
| enableAllEventHandlers()                 | Enable all event handlers.  |
| removeAllEventHandlers()                 | Removes all event handlers.   |
| setPlaybackHandler(event)                | Set playback handler for specified event. The possible <b>event</b> parameter values are: fsplayer.api.PlaybackController.PAUSE_PLAYBACK, fsplayer.api.PlaybackController.START_PLAYBACK, fsplayer.api.PlaybackController.ANIMATION_STEP_CHANGED, |

|                              |  |
|------------------------------|--|
|                              | fsplayer.api.PlaybackController.SLIDE_POSITION_CHANGED,<br>fsplayer.api.PlaybackController.CURRENT_SLIDE_INDEX_CHANGE<br>fsplayer.api.PlaybackController.SLIDE_LOADING_COMPLETE,<br>fsplayer.api.PlaybackController.PRESENTATION_PLAYBACK_COM  |
| removePlaybackHandler(event) | Set playback handler for specified event.<br>The possible <b>event</b> parameter values are:<br>fsplayer.api.PlaybackController.PAUSE_PLAYBACK,<br>fsplayer.api.PlaybackController.START_PLAYBACK,<br>fsplayer.api.PlaybackController.ANIMATION_STEP_CHANGED,<br>fsplayer.api.PlaybackController.SLIDE_POSITION_CHANGED,<br>fsplayer.api.PlaybackController.CURRENT_SLIDE_INDEX_CHANGE<br>fsplayer.api.PlaybackController.SLIDE_LOADING_COMPLETE,<br>fsplayer.api.PlaybackController.PRESENTATION_PLAYBACK_COM |

**See also**

[PlaybackControllerEvents Class](#)

[Player Class](#)

[JavaScript support in iSpring](#)

**1.6.4. fsplayer.api.PlaybackControllerEvents Class**

The **PlaybackControllerEvents** class exposes the following properties:

| Properties                   |        |   |
|------------------------------|--------|---|
| Property                     | Type   | Description   |
| PausePlaybackHandlers        | Object | Stores array of event handlers associated with pause playback event.          |
| StartPlaybackHandlers        | Object | Stores array of event handlers associated with start playback event.          |
| AnimationStepChangedHandlers | Object | Stores array of event handlers associated with animation step changing event. |
| SlidePositionChangedHandlers | Object | Stores array of event handlers associated with slide position changing event. |

|                                   |        |  |
|-----------------------------------|--------|--|
| CurrentSlideIndexChangedHandlers  | Object | Stores array of event handlers associated with slide index changing event.               |
| SlideLoadingCompleteHandlers      | Object | Stores array of event handlers associated with slide loading is completed event.         |
| PresentationPlaybackCompleteHandc | Object | Stores array of event handlers associated with presentation playback is completed event. |

**See also**

[PlaybackController Class](#)  
[JavaScript support in iSpring](#)

**1.6.5. fsplayer.api.PresentationInfo Class**

The **PresentationInfo** class stores the presentation information and provides the following properties:

| Methods         |  |
|-----------------|--|
| Method          | Description  |
| getTitle()      | Returns presentation title.  |
| getDuration()   | Returns presentation duration in seconds.  |
| getSlides()     | Returns the <a href="#">SlidesCollection</a> object providing the presentation content information.              |
| getPresenters() | Returns the <a href="#">PresentersCollection</a> object providing the information about presentation presenters. |

**See also**

[PresentersCollection Class](#)  
[SlidesCollection Class](#)

[JavaScript support in iSpring](#)

### 1.6.6. fsplayer.api.SlidesCollection Class

The **SlidesCollection** class stores the information about number of presenters in presentation:

| Methods             |  |
|---------------------|--|
| Method              | Description  |
| getSlidesCount()    | Returns number of slides in the presentation.  |
| getSlideInfo(index) | Returns <a href="#">SlideInfo</a> class which represents an individual presentation slide with the specified slide index.<br><b>Note:</b> Slide index is zero-based. |

**See also**

[PresentationInfo Class](#)

[JavaScript support in iSpring](#)

### 1.6.7. fsplayer.api.PresentersCollection Class

The **PresentersCollection** class stores the information about number of presenters in presentation:

| Methods    |   |
|------------|---|
| Method     | Description                                     |
| getCount() | Returns number of presenters in the collection. |

**See also**

[PresentationInfo Class](#)

[JavaScript support in iSpring](#)

### 1.6.8. fsplayer.api.SlideInfo Class

The **SlideInfo** class provides information about the particular slide of the presentation such as title, duration, text, etc. It exposes the following methods:

| Methods                      |  |
|------------------------------|--|
| Method                       | Description  |
| <code>getTitle()</code>      | Returns slide title.   |
| <code>getDuration()</code>   | Returns slide duration in seconds.   |
| <code>getStepsCount()</code> | Returns number of slide animation steps.                                       |
| <code>isLoading()</code>     | Returns Boolean value indicating whether the slide has been loaded completely. |

**See also**

[PresentationInfo Class](#)

[SlidesCollection Class](#)

[JavaScript support in iSpring](#)

**1.7. Support**

We are trying to develop our product to the maximum ease of use. But in case you have problems using iSpring, please first consult the documentation provided with the product and try to find the competent answers to your questions in the [iSpring FAQ](#) base.

If your question remains unanswered, please get in touch with us through [online feedback form](#). We will be happy to give you our helping hand and to find a quick solution to your request.

We appreciate any comments, concerns, kudos or critique. Share your opinion on the service you have received when using our product. Feel free to send us your feedback at [support page](#) on [www.ispringsolutions.com](http://www.ispringsolutions.com). Also now you can post your questions and opinions, exchange ideas and discuss technical issues on [iSpring Community Forum](#).

All users of iSpring are entitled to absolutely free and prompt technical support. We stop providing support only to refunded copies' users.

**Free Technical Support** includes:

- Unlimited free patches for stability-related bugs
- The iSpring online knowledge base
- Email questions that will be answered within five business days

To get the most benefit out of iSpring PRESENTER, sign up for a **Premium Technical Support** or [Consulting Service](#) and receive prompt and personalized service from the iSpring Support Team.

**Premium Technical Support** includes:

- Free iSpring upgrades including major releases and patches
- Unlimited telephone support
- Email questions answered within one business day
- 20% discount on iSpring Consulting services

[Contact us for details](#) about Premium Technical Support and Consulting Services.