

FSI VIEWER

Let your website visitors have a closer look

User Manual



Version 2017

FSI Viewer JS

FSI TouchZoom

FSI QuickZoom

FSI ImageFlow

FSI Showcase JS

FSI Pages JS

Developed by:

NeptuneLabs GmbH
Lagesche Str. 32
D-32657 Lemgo
Germany

© 2009-2017 NeptuneLabs. All rights reserved.

Last updated:

February 2017

FSI TouchZoom	17.02
FSI Viewer JS	17.02
FSI Pages JS	17.02
FSI ImageFlow	17.02
FSI QuickZoom	17.02
Manual Revision	055

All brands and product names are trademarks or registered trademarks of the respective producers.
FSI Viewer, FSI Server and NeptuneLabs are registered trademarks of NeptuneLabs GmbH, Germany.

Table of Contents

1 Introduction	8
2 FSI Viewer JS	10
2.1 How it works	11
2.2 Using FSI Viewer JS	12
2.2.1 The Menu Bar	13
2.2.2 Mouse Modes	13
2.2.3 Additional Buttons	14
2.2.4 Using the navigator window	15
2.3 Index structure of FSI Viewer JS	15
2.4 Configuration	15
2.4.1 Using XML Configuration Files (*.xml)	15
2.4.2 Hierarchy of Configuration Parameters	17
2.4.3 Retrieving Parameters from FSI Server	19
2.4.4 Using Custom Language Files	19
2.4.5 Enabling the Debug Mode	20
2.5 FSI Viewer JS Parameters	20
2.5.1 Global Parameters for Hot Spots	39
2.5.2 <camera> Section	42
2.5.3 <virtualSpace> Section	49
2.5.4 <hotspots3d> Section	51
2.5.5 <video> Section	63
2.5.6 Plugin Parameters	67
2.6 Creating and using Spin Videos	70
2.7 Embedding the viewer at runtime via JavaScript	71
2.8 JavaScript Interface	73
2.8.1 Public Methods	73
2.8.2 Callbacks	81
3 FSI TouchZoom	88
3.1 Initializing FSI TouchZoom manually	89
3.2 FSI TouchZoom Parameters	90
3.3 Initializing FSI TouchZoom manually	93
3.4 JavaScript Interface	95
4 FSI QuickZoom	98
4.1 How it works	99
4.2 FSI QuickZoom Parameters	99
4.3 Initializing FSI QuickZoom manually	105
4.4 JavaScript Interface	106

5 FSI ImageFlow	110
5.1 How it works	111
5.2 FSI ImageFlow Parameters	111
5.3 Embedding the viewer at runtime via JavaScript	126
5.4 JavaScript Interface	128
6 FSI Showcase JS	132
7 FSI Pages JS	134
7.1 How it works	134
7.2 Usage	134
7.2.1 Defining Image Collections	136
7.2.2 Aspect Ratio of the Pages	138
7.2.3 Printing Pages	138
7.2.4 Special URL Values	139
7.2.5 FSI Pages JS Presets in FSI Server Interface "Publish to Web"	140
7.3 FSI Pages JS Parameters	140
7.3.1 General Parameters	141
7.3.2 Layout/ Appearance Parameters	148
7.3.3 Link Parameters	160
7.3.4 Plug-in Parameters	165
7.4 JavaScript Interface	170
7.4.1 Public Methods	170
7.4.2 Callbacks	176
8 FSI Viewer	182
8.1 Compatibility and User System Requirements	182
8.2 Using FSI Viewer	182
8.3 The Menu Bar	183
8.4 Mouse Modes	183
8.5 Additional Buttons	184
8.6 Using the navigator window	185
8.7 Keyboard Shortcuts	185
8.8 Index structure of FSI Viewer	186
8.8.1 Licensing FSI Viewer	186
8.9 Integration into HTML Pages	187
8.9.1 Required HTML-Source Code	187
8.10 Configuration	188
8.10.1 Using XML Configuration Files (*.xml)	189
8.10.2 Using HTTP Queries	191
8.10.3 Hierarchy of Configuration Parameters	192
8.10.4 Default Configuration	193

8.10.5 Configuration Files	193
8.10.6 Passing Parameters by HTTP Query	194
8.10.7 Retrieving Parameters from Imaging Servers	194
8.10.8 Parameters	195
8.10.9 Applying Effects to Specific Images Only	218
8.10.10 Using Custom Language Files	226
8.11 Virtual 360° Presentations	226
8.11.1 Parameters for Virtual 360° Presentations	230
8.12 Debug Mode	236
8.12.1 Enabling the Debug Mode	236
8.12.2 Using the Debug Window	237
8.13 Automated Implementation of Images	238
8.14 Plug-ins	241
8.14.1 Deactivating Plug-ins	242
8.14.2 Defining Plug-in Parameters	243
8.14.3 Defining Plug-in Parameters dynamically	243
9 FSI Pages Add-on	246
9.1 Using FSI Pages	246
9.2 FSI Pages Converter	247
9.3 Aspect Ratio of the Pages	247
9.4 Layout and Skins	248
9.5 Cropping	249
9.6 Defining Image Collections	250
9.7 Page Reading Order	252
9.8 Printing Pages	253
9.9 Saving Pages	254
9.10 Saving PDF or other document types	254
9.11 Searching in FSI Pages	256
9.12 Hyperlinks	258
9.13 Publishing PDF Documents with Links using FSI Pages Converter and FSI Server	259
9.14 Providing XML page data to FSI Pages	259
9.15 Link Parameters and Hierarchy of Link Parameters	265
9.16 Relative and Absolute Links	267
9.17 Special URL Values	268
9.18 Modifying Link Values at Runtime	269
9.19 Page Overlays	270
9.20 FSI Pages Presets in FSI Server Interface "Publish to Web"	271
9.21 Event Notifications and Actions	272
9.22 Parameters to retrieve Image Collections	274

9.23 Basic Parameters	277
9.24 Parameters for the Front- and Backcover	284
9.25 Parameters defining Layout and Appearance	287
9.26 Parameters for Links on Pages	297
9.27 Parameters for FSI Pages Thumbnails	303
9.28 Parameters for Statistics	307
9.29 Advanced Parameters	309
10.1 FSI Pages mobile	328
10.1.1 Pages Mobile Parameters	329
10.1.2 Pages Mobile Plug-in Attribute targetdevice	331
11 FSI Showcase Add-on	332
11.1 Using FSI Showcase	332
11.2 Defining Image Lists	332
11.3 FSI Showcase Layout	335
11.4 Showcase Parameters	337
11.5 Basic Parameters	338
11.6 ImageList Parameters	344
11.7 Advanced Parameters	346
12 Plug-in Reference	362
12.1 AddThis	363
12.2 BackgroundImage	369
12.3 Chapters	371
12.4 ClockProgress	376
12.5 ColorAdjust	380
12.6 ContextMenu	383
12.7 CustomButton	385
12.8 Fullscreen	390
12.9 History	392
12.10 HotSpots	394
12.11 Imprint	408
12.12 JavaScript Bridge	411
12.13 LargeToolTips	424
12.14 Magnifier	429
12.15 MaxZoom	432
12.16 Measure	433
12.17 Mousemodes	442
12.18 MousemodeSelect	444
12.19 Music	446
12.20 NavExtension	449

12.21 Notepad	451
12.22 PagesLibrary	463
12.23 PagesMirror	467
12.24 PageSounds	469
12.25 PagesThumbBar	470
12.26 PrintSave	478
12.27 Resize	482
12.28 SelectFrame	484
12.29 ShoppingList	489
12.30 SoftwareCursor	495
12.31 StickyNotes	496
12.32 Synchronize	501
12.33 ZoomMeter	503
13 Appendix	506
A Escaping or URL-encoding parameter values	506
B HTML Tags available in FSI Viewer	506
C Example of a _default.xml file for FSI Viewer Flash	507
D Example of an image specific FSI configuration file	507
E Example of a FSI Viewer JS skin css file	508
13 Appendix	508
F Escaping or URL-encoding parameter values	508
G HTML Tags available in FSI Viewer	509
H Example of a custom skin CSS file	509
Index	515

1 Introduction

The JavaScript based viewers FSI Viewer JS, FSI TouchZoom, FSI QuickZoom, FSI Showcase JS and FSI ImageFlow bring outstanding zoom qualities to almost all devices. They are highly customizable and work independently from FSI Viewer Flash.

FSI Viewer Flash and its Add-ons FSI Pages and FSI Showcase are based on Adobe Flash and have been designed to display high resolution images on the internet requiring low bandwidth only.

FSI Server is the basis of all viewers which request the image data from it. By using FSI Server, only one high resolution source image is required for each image to be displayed.

Thank you for using NeptuneLabs software!

www.neptunelabs.com : Online Resources for FSI Viewer

Visit www.neptunelabs.com for software updates, regularly updated samples, tutorials and downloads.

2 FSI Viewer JS

FSI Viewer JS is a JavaScript based tool to display 2D image zoom and 360° Object spins on desktop computers and mobile devices. Depending on the device capabilities it offers in place zoom or pinch zoom support. FSI Viewer JS also supports optional multiple axis spin.

Using an optional video, the rotation of the 360° object can be displayed extremely smooth on devices that support video capturing. Optional hot spots with HTML content can be added by providing 3D coordinates (x,y,z) in millimeters.

Imaging Servers

FSI products covered by this manual can at present securely be run with FSI Server from version 2016.

Supported devices

- iOS 4 or higher
- Android 4.x based devices
- Windows Phone 8 and Windows RT devices
- Windows 8 with touch enabled browsers
- Mouse zoom: desktop or laptop computers & all major browsers

Usage

I) add the fsviewer.js script to the head of your document:

Adding the script

```
<head>
[...]
//adding the fsviewer.js script
<script src="viewer/applications/viewer/js/fsviewer.js"
type="text/javascript"></script>
</head>
```

II) Add a <fsi-viewer> tag to the part of your document body where the zoom and/or the 360° product spin should be displayed. Add the desired dimension and the imagesource/ directory to it:

Integrate FSI Viewer JS

```
<body>
[...]
<fsi-viewer width="500" height="600" src="/images/foo.tiff">

//here you can place the fallback content in case the browser
// cannot display the viewer, e,g,:
<b>Sorry</b>, your browser is not capable of running FSI Viewer
JS

</fsi-viewer>
```

Parameters are added to the `<fsi-viewer>` tag. Note that you can as well use FSI Viewer configuration files (required when using hot spots):

Adding Parameters to the viewer

```
<head>
[...]
//adding the fsiviewer.js script
<script src="js/fsiviewer.js" type="text/javascript"></script>
</head>

<body>
[...]

//Adding parameters directly to the tag:
<fsi-viewer width="500" height="600" src="/images/sample.jpg"
debug="true" skin="silver">
</fsi-viewer>

//Example with using a FSI Viewer configuration file:
<fsi-viewer width="500" height="600" src="/images/sample.jpg"
cfg="sample/sample_configuration">
</fsi-viewer>
```

2.1 How it works

The script uses the given `<fsi-viewer>` tag, which is similar to an `` tag, to display a single image zoom and/or 360° object spin.

If a video has been defined, the script captures video frames on start to provide an ultra smooth rotation.

Without a video (or on devices not supporting video capturing) a fixed number of images from different viewing angles will be used to display the object spin. The user can drag the image left and right to spin and zoom in/out either using pinch zoom or by clicking on the object.

Note on Internet Explorer:

In order to make FSI Viewer JS work, Internet Explorer must run in IE8 quirks, IE9 or higher quirks or standards mode. You can ensure that by adding an X-UA-Compatible meta tag to the head section of your web page, for example:

```
<meta http-equiv="X-UA-Compatible" content="IE=Edge">
```

2.2 Using FSI Viewer JS

FSI Viewer JS is navigated by the menu bar (here seen below the image) and by using the mouse directly on the image. For example, you can select a section of the image you want to magnify. The mouse functions are determined by the corresponding buttons on the menu bar (zoom, pan, rotate, etc.).

The optional small navigator window (bottom right) displays the position of the image section currently viewed.






2.2.1 The Menu Bar







The menu bar might look different depending on skins or additional plugins.

Main Functions



	<p>Back to Initial View. Undoes zoom, pan and rotation and restores the initial view. (identical with pressing space bar)</p>
	<p>Zoom In</p>
	<p>Zoom Out</p>

2.2.2 Mouse Modes

The `→ Mousemodes` or `→ MousemodeSelect` plug-in is required to display the following mouse mode buttons.

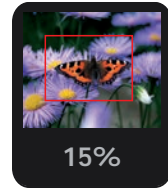
	<p>Mouse - Zoom In Choose this function to enlarge segments using the mouse. Click on the image and drag the frame over the desired segment. Alternatively you can click on the image, without marking a segment. The image will then be magnified in steps. To zoom out in steps, hold down the CTRL-key and click on the image.</p>
	<p>Mouse - Pan Choose this function to pan the image using the mouse. Click on the image and drag in the desired direction. To return to the original view, hold down the CTRL-key and click on the image.</p>
	<p>Mouse - Turn Choose this function to rotate the image around the z-axis. Click on the image and drag in whatever direction you want to turn the object (to the right or the left). To reset the rotation, hold down the CTRL-key and click on the image.</p>
	<p>Mouse - Rotate (only for 360° presentations) Choose this function to rotate the object around the y-axis or the x-axis if available. Click on the image and drag to the left or to the right. Move the mouse up or down to rotate the object around the x-axis. To reset rotation, hold down the CTRL-key and click on the image.</p>

2.2.3 Additional Buttons

	<p>Hide / Display menu Displays or hides the user interface.</p>
	<p>Information Displays information about the viewer (optional, can be hidden by altering the skin CSS)</p>

2.2.4 Using the navigator window

In the navigator window you can see a miniature presentation of the entire image. The segment which is currently viewed is framed. In the navigator window you can change the current segment either by dragging the frame or by clicking on the desired area of the image.



2.3 Index structure of FSI Viewer JS

The index structure of the JS viewers as found in `WEBAPPS/fsi/viewer/`:

Directory	Content Description
applications	contains the .js scripts for the according viewers
config	contains the configuration files
languages	User interface language files
plugins	FSI Plug-ins extending FSI Viewer at runtime
skins	FSI Skins containing the visible parts of the user interface

Please Note: You need to make sure that your web server properly serves files with the following extensions: *.js, *.xml

2.4 Configuration

2.4.1 Using XML Configuration Files (*.xml)

FSI Viewer JS uses XML configuration files which can easily be edited with any text editor.

The default configuration file "`_default.xml`" - which is located in `WEBAPPS/fsi/viewer/` is the most important XML configuration file and will always be evaluated first.

To use FSI Viewer with additional configuration files you have to place the file in the `WEBAPPS/fsi/viewer/config/` folder and provide the name of the configuration file using the "`cfg`" parameter in the code:

Using FSI Viewer Configuration Files

```
<head>
[...]
//adding the fsiviewer.js script
<script src="js/fsiviewer.js"
type="text/javascript"></script>
</head>

<body>
[...]
```

//Example with using a FSI Viewer configuration file:

```
<fsi-viewer width="500" height="600"
src="/images/sample.jpg" cfg="sample/sample_configuration">
</fsi-viewer>
```

Configuration files are structured into configuration groups (XML nodes) containing parameter names and values. Only parameters in these groups will be recognized by FSI Viewer. Configuration files do not need to contain all groups though. Each parameter has to be provided as an individual XML child node of a group where the node name is the parameter name and the parameter value needs to be provided as "value" attribute of the node.

Structure of XML configuration files

```
<fsi_parameter>
<image>
  <parameter value="" />
</image>

<plugins>
  <parameter value="" />
</plugins>

<options>
  <parameter value="" />
</options>

</fsi_parameter>
```


Simple XML configuration file

```
<fsi_parameter>
  <image>
    <path value="samples/Watch.jpg" />
  </image>
  <options>
    <skin value="silver" /><
    debug value="1" />
  </options>
</fsi_parameter>
```

Please keep in mind the following rules when using XML configuration files:

- attribute values must be provided in double quotes
- the last occurrence of a given parameter will be used
- configuration files might be cached by your browser, so that changes in these files require flushing the browser cache. This does not apply when using the debug mode.
- make sure to save configuration files in UTF-8 format if your configuration file contains language specific diacritics, Japanese characters and alike

You can comment on or disable sections of an XML configuration file using the usual XML comment syntax:

Comments and disabling parameters temporarily

```
<image>
  <!-- This is a comment -->
  <path value="image.tif" />

  <!-- The following section will be ignored -->
  <!--
  <path value="anotherimage.tif" />
  -->
</image>
```

2.4.2 Hierarchy of Configuration Parameters

As explained before parameters can be provided in two different ways:

1. The default configuration file "_ default.xml" - located in WEBAPPS/fsi/viewer/ - will always be evaluated first

2. You can store image specific parameters in additional *.xml files in WEBAPPS/fsi/viewer/config/

Parameters specified in multiple ways – for example in the default configuration and by custom configuration files– will be evaluated in the following way:

- Parameters defined in configuration files override parameters defined in the default configuration file "_default.xml".

The list below shows the hierarchy resulting from the sequential evaluation of the different configuration methods:

Configuration Hierarchy

1. Configuration file (*.xml)
2. Default configuration file (_default.xml)

Choosing the adequate Configuration Method

So far you learned that FSI Viewer JS can be configured by any combination of:

- The default configuration file (_default.xml)
- Additional configuration files (*.xml)

These configuration options are not only alternate methods. Combining these methods offers the possibility to provide configuration parameters as easily as possible and as dynamically as required. The following considerations might be helpful when deciding which method to choose for a certain parameter.

Default Configuration

You can easily change global parameters that apply to all FSI Viewer instances by editing a single file. This is therefore the recommended configuration method for:

General configuration options like:

- Skin
- Language
- MenuAlign, AnimationSpeed and alike

Configuration Files

Separate configuration files should be used for parameters specific to a group of images or a specific way to display images. You might for example want to setup individual configuration files for different types of FSI Viewer JSinstances, like "small_no_interface.xml", "large_with_hotspots.xml" as an example.

Usually you don't need to create a configuration file for individual images, as you can specify the image path or other unique parameters dynamically in the code.

2.4.3 Retrieving Parameters from FSI Server

FSI Viewer JS is capable of retrieving image specific parameters from FSI Server dynamically. This way you do not need to define these parameters explicitly. The list below shows some typical parameters that can be retrieved from FSI Server automatically:

- **ImageWidth**
- **ImageHeight**
- **ImageTilesX**
- **ImageTilesY**
- **SceneSets**
- **Image Collections**
- **Hyperlinks**
- **Hotspots**

In which case will these values be retrieved automatically?

FSI ViewerJS tries to retrieve the parameters ImageWidth and ImageHeight from the server if one of these values has not been defined manually. If the server delivers additional parameters (e.g. ImageTilesX, ImageTilesY) these parameters will only have an effect if they have not been defined manually.

2.4.4 Using Custom Language Files

In case you want to edit tool tips for existing languages or add support for additional languages you can edit or create language files in the subdirectory "/languages" of your FSI Viewer setup directory.

Each language file contains a list of phrases used in FSI ViewerJS user interface. The language files are XML files and need to be saved in UTF-8 format. The attribute "id" specifies the context of each phrase. Please note that you need to use XML entities for characters not allowed in XML files, like e.g.

&	for "&"
<	for "<"
>	for ">"
"	for "
'	for ,

You can use basic HTML-tags to apply text formatting to the tool tips.

Excerpt from a language file

```

<Data language="english" color="000000" font="_ sans"
offsetTop="0" offsetLeft="0"
offsetWidth="0" offsetHeight="0">

    <tip id="ZoomOut"><b>Zoom Out</b><br/>Click to
    decrease magnification</tip>
    <tip id="ZoomIn"><b>Zoom In</b><br/>
    Click to increase
    magnification</tip>
    <tip id="Downloading Data">Downloading:</tip>
[...]
```

You can duplicate existing language files or create new files as needed. To use custom language files, please use the "language" parameter and set the parameter value to the filename without the ".xml" extension.

2.4.5 Enabling the Debug Mode

The debug mode can be enabled or disabled by passing the "Debug" parameter either in the code snip or in an XML configuration file.

Please use "debug=1" or "debug=true" to enable the debug mode and "debug=0" (or omit the debug parameter) to disable the debug mode. To disable the debug mode permanently you can safely delete the "/debug" directory of your FSI setup directory.

Example:

If FSI Viewer JS does not work or look as expected in an HTML page you can quickly debug FSI Viewer as follows:

- Open the HTML page containing FSI Viewer JS
- Open the "Inspect Element" feature of your browser
- in the "Console" part you can now check if everything is implemented and loaded correctly

2.5 FSI Viewer JS Parameters

You can optionally modify the way FSI Viewer JS works by passing parameters to the script. To do so, pass an object containing the parameters and value as an

object to the constructor:

Example:

```
<head>
[...]
//adding the fsiviewer.js script
<script src="js/fsiviewer.js" type="text/javascript"></script>
</head>

<body>
[...]

//Adding parameters directly to the tag:
<fsi-viewer width="500" height="600" src="/images/sample.jpg"
debug="true" skin="silver">
</fsi-viewer>
```

debug	
Description	display debug and status information in the JavaScript console
Syntax	Boolean
Default	false
Context	FSI Viewer JS

Display debug and status information in the browser's JavaScript console.

headers	
Description	display debug and status information in the JavaScript console
Syntax	String
Default	"cors"
Context	FSI Viewer JS

The "&headers=" parameter to use for image requests.
The default value "cors" ensures cors compatibility for cross-origin scenarios.

autoSpinSpeed	
Description	Time in seconds for a full rotation
Syntax	float
Default	-
Context	FSI Viewer JS

Time in seconds for a full rotation. Negative values reverse spin direction.

cfg	
Description	relative path to an XML configuration file in FSI Viewer format
Syntax	String
Default	""
Context	FSI Viewer JS

The relative path to an XML configuration file in FSI Viewer format. When using a relative path the location specified by the FSI Viewer "FSIBase" parameter will be used, which is [viewer]/config/ by default.

hideUI	
Description	hides the user interface
Syntax	string
Default	false
Context	FSI Viewer JS

Hides or displays the user interface.

Possible values:

true

false

onMobileDevices: UI is hidden on mobile devices

ifNoInPlaceZoom: UI is hidden if in place zoom is deactivated

noNav	
Description	hides the navigation window
Syntax	boolean
Default	false
Context	FSI Viewer JS

Hides or displays the navigation window.

autoHideNav	
Description	automatically hides the navigation window
Syntax	boolean
Default	false
Context	FSI Viewer JS

Hides or displays the navigation window automatically if the magnification is at minimum.

navWidth	
Description	defines the width of the navigation frame
Syntax	String
Default	"120"
Context	FSI Viewer JS

Defines the width of the navigation frame displayed in the right corner on the bottom.

navHeight	
Description	defines the height of the navigation frame
Syntax	String
Default	"120"
Context	FSI Viewer JS

Defines the height of the navigation frame displayed in the right corner on the bottom.

progressBarWidth	
Description	defines the width of the progress bar
Syntax	float
Default	35
Context	FSI Viewer JS

Defines the width of the progress bar in percent of the viewport width.

progressBarHeight	
Description	defines the height of the progress bar
Syntax	float
Default	14
Context	FSI Viewer JS

Defines the height of the progress bar in pixel.

progressBarHAlign	
Description	defines the horizontal position of the progress bar
Syntax	string
Default	"center"
Context	FSI Viewer JS

Defines the horizontal position of the progress bar, possible values: left, center, right.

progressBarVAlign	
Description	defines the vertical position of the progress bar
Syntax	string
Default	"bottom"
Context	FSI Viewer JS

Defines the horizontal position of the progress bar, possible values: top, middle(or center), bottom.

urlLoadingAnimation	
Description	defines the loading animation shown
Syntax	string
Default	"/skins/resources/loading.svg"
Context	FSI Viewer JS

Defines the loading animation shown. Custom animations can be placed here: /skins/resources/. Setting the parameter to "false" discards the display.

skin	
Description	defines the skin to display
Syntax	string
Default	"black"
Context	FSI Viewer JS

Defines the skin which will be displayed. FSI Viewer JS comes with three default skins, "black", "white" and "silver".

Skins are defined via CSS and can be found here: [WEBAPPS/fsi/viewer/skins](#).

It is also possible to alter existing skins or to define custom skins in this location.

In order to use a custom skin, the easiest way would be to duplicate one of the default skin CSS files and adapt them to your liking. make sure to use a custom class in front af ALL classes defined in the custom CSS,, otherwise you will break other skins on the same HTML document.

General appearance CSS rules, which are valid on an overall level, (e.g. the logo container, height of the menubar, etc.) are defined in the corresponding core CSS files, which can be found in WEBAPPS/fsi/viewer/skins/resources. (fsi-core.css, fsi-viewer-core.css)

IMPORTANT: Never change the classes of the core CSS in the core files directly. If you would like to adapt the CSS rules listed there, please overwrite them in a custom skin file or in the HTML document instead.

You can find detailed tutorials for changing the appearance of the skin in the *support section of our website*.

For a detailed example of a basic custom skin, please go to → *"Example of a custom skin CSS file"*.

enableZoom	
Description	enable the user to zoom in and out
Syntax	Boolean
Default	true
Context	FSI Viewer JS

Enable the user to zoom in and out.

On touch enabled devices the zoom is based on the native pinch zoom gesture. On non touch enabled devices, in place zoom by clicking the object or dragging up/down will be enabled instead. Alternatively, the menu bar can be used for zooming.

desktopClickZoomScale	
Description	the magnification level to use when clicking the object
Syntax	Float
Default	3
Context	FSI Viewer JS

For desktop browsers only. The magnification level to use when clicking the object.

dir	
Description	path to the source images for the object spin on FSI Server
Syntax	String
Default	""
Context	FSI Viewer JS

The path to the source images for the object spin on FSI Server.
The viewer will load the list of images from the given directory.

imagesrc	
Description	path to the single images used for FSI Viewer JS
Syntax	String
Default	""
Context	FSI Viewer JS

Defines the path to single images used for FSI Viewer JS.

imagesources	
Description	paths to single images used for FSI Viewer JS
Syntax	String
Default	""
Context	FSI Viewer JS

Defines the paths to single images used for FSI Viewer JS. Example: "dir1/1.jpeg, dir1/2.jpeg, dir2/1.jpeg,..."

backgroundColor	
Description	the background color to use
Syntax	6-digit hex-color
Default	#FFFFFF
Context	FSI Viewer JS

The background color to use. The color should match the video's or image's background color (usually white or black).

mouseSensitivity	
Description	sensitivity of the mouse movement when rotating the object
Syntax	float
Default	1.0
Context	FSI Viewer JS

Sensitivity of the mouse (finger) movement when rotating the object.

pellets	
Description	viewer draws optionally small dots on the video
Syntax	boolean
Default	true
Context	FSI Viewer JS

When using a spin video you can optionally have the viewer draw small dots on the video. This improves the visual impression of the video in higher resolutions.

pelletsOpacity	
Description	defines opacity of the dots ("pellets") drawn on the video
Syntax	float
Default	0.09
Context	FSI Viewer JS

Opacity of the dots ("pellets") drawn on the video, if the pellets parameter is enabled.

drawCube	
Description	draws a 3D cube wireframe representing the spin object space
Syntax	boolean
Default	false
Context	FSI Viewer JS

Draws a 3D cube wireframe representing the object space of the spin as defined by the "virtualSpace" parameters.

The drawCube option can be useful to validate the "virtualSpace" parameters.

This option only works on desktop computers using the HTML canvas element.

drawTable	
Description	draws a virtual rotating table plate
Syntax	boolean
Default	false
Context	FSI Viewer JS

Draws a virtual rotating table plate. **This option only works on desktop computers using the HTML canvas element.**

tableDiameter	
Description	defines diameter of virtual rotating table plate
Syntax	Float
Default	950
Context	FSI Viewer JS

Defines the diameter of the virtual rotating table plate in mm. **This option only works on desktop computers using the HTML canvas element.**

drawGroundPlate	
Description	draws a virtual ground plate
Syntax	boolean
Default	false
Context	FSI Viewer JS

Draws a virtual ground plate.

This option only works on desktop computers using the HTML canvas element.

effects	
Description:	Image manipulation parameters for all images
Syntax:	String
Default:	—
Context:	FSI Viewer JS

The effects parameter can be used to pass image manipulation parameters to the imaging server. Typical manipulation parameters include the image saturation ("saturation") and image sharpening ("sharpen"). If no effect is specified, the default settings of the server (profile) apply. Please refer to your FSI Server documentation for a list of available effect parameters.

initialView	
Description	defines the initial image segment
Syntax	String
Default	""
Context	FSI Viewer JS

Image section (and rotation) to display on startup.
The first two parameters are required as they specify the scene set and the scene.
For 2D images both values default to 1.

initialViewPersistent	
Description	Keep the InitialView as default view
Syntax	boolean
Default	false
Context	FSI Viewer JS

If this parameter is activated, clicking the "Reset" button will display the image section defined by the InitialView parameter instead of the entire image. If this parameter is set to "false", the initial view will be displayed on startup, but pressing the Reset button will reset the view to the entire image.

showPreviewWhileLoading	
Description	shows/hides preview images for spins while loading
Syntax	boolean
Default	true
Context	FSI Viewer JS

If this parameter is set to false, the preview images which are shown while the spin is loaded are hidden.

spinPreviewWhileLoading	
Description	shows/hides preview images for spins while loading
Syntax	boolean
Default	true
Context	FSI Viewer JS

If this parameter is set to false, a static image will be shown instead of the spin while loading. If `initialView` is defined, the image defined there will be shown.

showProgressbar	
Description	shows/hides progress bar while loading
Syntax	boolean
Default	true
Context	FSI Viewer JS

If this parameter is set to false, the progress bar which is shown while the spin is loaded is hidden.

panLimits	
Description	determines how far the user is allowed to pan
Syntax	String
Default	"strict"
Context	FSI Viewer JS

Determines how far the user is allowed to pan. Possible settings are: "loose", "medium", "strict"

preventImageTransparency	
Description	prevents the image transparency
Syntax	Boolean
Default	false
Context	FSI Viewer JS

Prevents using the image transparency if a transparency channel exists.

useDevicePixelRatio	
Description	Use the device's pixel ratio to display even sharper images.
Syntax	Boolean
Default	true
Context	FSI Viewer JS

Use the device's pixel ratio to automatically display even sharper images.

adaptiveUISize	
Description	determines if interface scales according to the zoom level .
Syntax	Boolean
Default	true
Context	FSI Viewer JS

When set to true, the interface scales according to the zoom level.

inPlaceZoom	
Description	activates zooming in the FSI Viewer JS instance
Syntax	string
Default	"true"
Context	FSI Viewer JS

The parameter defines if in-place-zoom (zooming directly in the FSI Viewer JS instance) is activated.

Possible values:

true

false

auto: in-place-zoom is activated, but not on mobile devices

documentZoom	
Description	allows pinch-zoom on FSI Viewer JS instance
Syntax	string
Default	"auto"
Context	FSI Viewer JS

Activates pinch document zoom on FSI Viewer instance.

Possible values:

true

false

auto: pinch document zoom is activated on all touch-enabled devices

autoDisablePointerActions	
Description	disables events if viewer is larger than viewport
Syntax	Boolean
Default	true
Context	FSI Viewer JS

Disables mouse and touch events if viewer's height OR width is larger than the viewport width or height.

loadExactPreviewImagesAfterResize	
Description	Reload images in the new required resolution after a resize event
Syntax	Boolean
Default	"true"
Context	FSI Viewer JS

Reload images in the new required resolution after a resize event.

maxZoom	
Description	defines the maximum magnification level
Syntax	Float
Default	100
Context	FSI Viewer JS

Defines the maximum magnification level.

minInstanceSize	
Description	defines the minimum instance size
Syntax	Integer
Default	60
Context	FSI Viewer JS

Defines the minimum instance size of the viewer.

initialMouseMode	
Description	Mouse mode on startup
Syntax	String
Default	"auto"
Context	FSI Viewer JS

Specifies the selected mouse mode on startup.

Possible values:

- 0: (Zoom)
- 1: (Pan)
- 2: (Spin 360°)

menuButtonOrder	
Description	defines the button order of the menu bar buttons
Syntax	String
Default	-
Context	FSI Viewer JS

Defines the button order of the menu bar buttons. The buttons need to be listed in a string with the corresponding button IDs.

The following button IDs currently exist:

Reset, ZoomOut, ZoomIn, MouseMode_0, MouseMode_1, MouseMode_2, MaxZoom, HotSpots, ToggleAutoSpin, ToggleFullScreen

autoDestroy	
Description	automatically destroy instances
Syntax	boolean
Default	true
Context	FSI Viewer JS

Automatically destroy instances created with the fsi-viewer tag upon removing the tag from the document DOM.

spinDirectionImage	
Description	defines direction of rotation
Syntax	String
Default	"left"
Context	FSI Viewer JS

Specifies the direction of the spin rotation.

SceneSets	
Description	Scene sequence definition
Syntax	String
Default	1 – n tiles
Context	FSI Viewer JS

Defines the minimum instance size of the viewer.

Using this parameter you can define scene sets for 360° views. Scene sets define the order of scenes to be displayed.

The default is a single scene set containing all scenes (image tiles) from scene 1 to the last scene image. The default sequence is left to right, top to bottom in a tile image and the sequence of the <image> nodes for 360° views based on individual source images.

In the viewer scenes can be selected by moving the mouse horizontally and scene sets can be selected by moving the mouse vertically.

Syntax:

Individual sets have to be separated with ";" (semicolon).

Scenes within the sets are separated with "," (comma) and specify the 1-based index of the image from left to right, top to bottom up to the total number of scenes (TilesX x TilesY or individual source images).

Instead of individual scenes you can also specify scene ranges, e.g. "5-10".

If you leave out the start or the end of a scene range, 1 or respectively the last scene will be assumed in this case. If you specify a range using "->" the scenes from

the first number to the last scene image plus the first scene image up to the second number will be assumed.

Example:

```
SceneSets="1-12;13-24;25-36;37-48"
```

reverseSceneSets	
Description	reverses the Scene Sets
Syntax	boolean
Default	false
Context	FSI Viewer JS

Reverses the order of the Scene Sets.

NoSetLoop	
Description	No 360° rotation around the y-axis
Syntax	Boolean
Default	false
Context	FSI Viewer JS

The first and the last scene set will be considered ending points. (E.g. for partial rotations < 360°).

Please note: The blur animation is turned off if NoSetLoop is activated.

NoSceneLoop	
Description	No 360° rotation around the x-axis
Syntax	Boolean
Default	false
Context	FSI Viewer JS

The first and last scene in the each scene set will be considered ending points. (E.g. for partial rotations < 360°).

Please note: The blur animation is turned off if NoSceneLoop is activated.

2.5.1 Global Parameters for Hot Spots

enableHotspots	
Description	enables or disables hot spots
Syntax	boolean
Default	true
Context	FSI Viewer JS

You can set this value to false if hot spots have been defined, but shall not be displayed.

hotspotCallbackFunction	
Description	callback function for hot spots
Syntax	string/ function
Default	""
Context	FSI Viewer JS

The name of a custom JavaScript callback function in window scope to call each time a hot spot mouse action occurs.

Alternatively you can pass the function directly as a JavaScript parameter.

Example of a callback function prototype:

Callback Function Prototype

```
function onFSIViewerHSAction(strAction, viewer, idViewer,
idHotspot, node, param, angle)
{
  if (strAction == "click")
  {
    alert("clicked on hot spot " + idHotspot);
  }
}
```

Attribute	Description
action	a string representing the action "mouseover", "mouseout", "mousedown" or "click"
viewer	the FSI Viewer JS viewer instance calling the method
idViewer	the value of the "id" attribute of the viewer's <div> tag
idHotspot	the 0-based id of the hotspot
node	the <div> node containing the hot spot content
param	the content of the "actionParameter" parameter of the hot spot
angle	the hot spot's perimeterAngle value (the angle at which it's visible best)

hotspotLineColor1	
Description	lighter color of the hot spot marker lines
Syntax	6-digit hex color
Default	#FFFF00
Context	FSI Viewer JS

The lighter color of the hot spot marker lines.

hotspotLineColor2	
Description	darker color of the hot spot marker lines
Syntax	6-digit hex color
Default	#000000
Context	FSI Viewer JS

The darker color of the hot spot marker lines.

hotspotDotColor1	
Description	lighter color of the hot spot marker lines
Syntax	6-digit hex color
Default	#FFFF00
Context	FSI Viewer JS

The lighter color of the hot spot marker circle's gradient.

hotspotDotColor2	
Description	darker color of the hot spot marker lines
Syntax	6-digit hex color
Default	#000000
Context	FSI Viewer JS

The darker color of the hot spot marker circle's gradient

decodeHTMLEntitiesInHotSpotTexts	
Description	enables html entities in the hotspots text
Syntax	6-digit hex color
Default	#000000
Context	FSI Viewer JS

When enabled, basic HTML entities will be enabled in the hotspot text.

hotspotDotRadius	
Description	radius of the hot spot marker circle in px
Syntax	float
Default	4.0
Context	FSI Viewer JS

The radius of the hot spot marker circle in px.

2.5.2 <camera> Section

The camera section contains parameters describing the technical specifications and the position of the camera when shooting the 360° spin.

You only need to specify these values if you are using the virtual space, e.g. for hot spots.

For 2D zoom and spins without hot spots you can omit this parameter group.

digitalCropFactor	
Description	depends on the size of the optical sensor chip of the camera
Syntax	float
Default	1.0
Context	FSI Viewer JS

The value depends on the size of the optical sensor chip of the camera. For full frame cameras the value is "1.0". Common values are between 1.5 to 1.6. The crop factor is the same factor that affects the focal length of your camera's objective. E.g. the focal length of an objective with a full frame camera is 50mm. With a camera that has a crop factor of 1.5 the effective focal length would be $1.5 \times 50\text{mm} = 75\text{mm}$.

Please note that for an accurate calculation the rounded figure (e.g. 1.5) is not precise enough, since two to four decimal places are required.

In order to ensure an exact smooth video and hotspot implementation, please do the following calculation for specific sensor dimensions of the camera used. The size of the optical sensor chip can be found in the cameras technical specifications.

$$\frac{\sqrt{36^2 + 24^2}}{\sqrt{x^2 + y^2}} = \text{digitalCropFactor}$$

$36^2 + 24^2 =$ Reference value
(35mm "full frame" format; 36mm x 24mm,
crop factor: 1.0)

$x^2 + y^2 =$ Sensor dimensions of your camera
(width and height)

Please find below the calculation for a Canon 7D camera (which has the sensor dimensions of 22.3mm x 14.9mm) as an example:

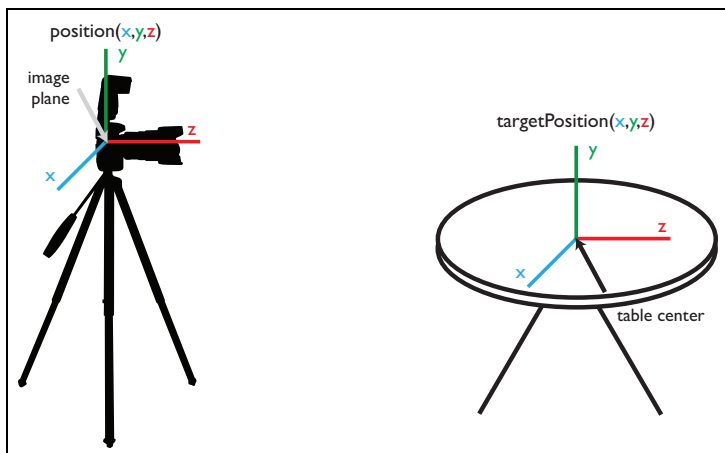
$$\frac{\sqrt{36^2 + 24^2}}{\sqrt{22.3^2 + 14.9^2}} = 1.6132$$

focalLength	
Description	focal length of the camera objective
Syntax	float (in mm)
Default	50
Context	FSI Viewer JS

The focal length of the camera objective.

position, targetPosition	
Description	defines the position of the camera and the object
Syntax	float (in mm)
Default	
Context	FSI Viewer JS

position (x,y,z) and targetPosition (x,y,z) can be used instead of distanceToTableCenter, heightAboveTable and targetHeightOffset. (see illustration below)



Explanation position(x,y,z)	
x	if image plane is in line with table center, x is 0
y	height of the image plane in reference to the table center
z	the distance between image plane and table center

Explanation targetPosition(x,y,z)	
x	if target is in table center, x is 0
y	height of the target in reference to the table center
z	the distance between target and table center

distanceToTableCenter	
Description	distance from cameras image plane to the center of the turntable
Syntax	float (in mm)
Default	0
Context	FSI Viewer JS

The straight distance from the cameras image plane to the center of the turn table in mm (see illustration below).

heightCamera	
Description	distance from image plane of the camera to the ground
Syntax	float (in mm)
Default	0
Context	FSI Viewer JS

The straight distance from the optical center of your camera to the ground in mm (see illustration below).

heightTable	
Description	distance from top of the turn table to the ground
Syntax	float (in mm)
Default	0
Context	FSI Viewer JS

The straight distance from the top of the turn table to the ground in mm. (see illustration below)

targetHeightOffset	
Description	height of the camera target spot at the center of the turntable
Syntax	float (in mm)
Default	0
Context	FSI Viewer JS

The height of the camera target spot at the center of the turn table in mm. If you vertically target the table center the value is 0.0 mm. (see illustration below)

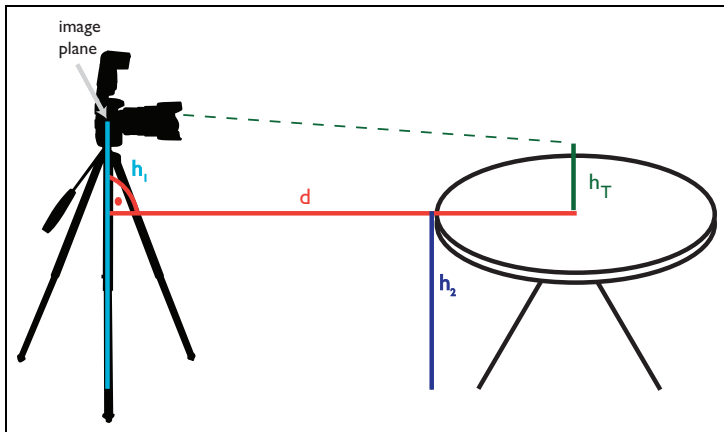
targetWidthOffset	
Description	horizontal offset of the camera target spot
Syntax	float (in mm)
Default	0
Context	FSI Viewer JS

The horizontal offset of the camera target spot at the vertical the center of the turn table related to the center of the turn table in mm. If you target the horizontal table center (default) the value is 0.0 mm.

heightAboveTable	
Description	height of the image plane above the table
Syntax	float (in mm)
Default	500
Context	FSI Viewer JS

Alternative parameter if heightCamera and heightTable are unknown. Only applies if both parameters are undefined.

Camera Positioning Illustration



Attribute	Description
hT	Parameter: targetHeightOffset
h1	Parameter: heightCamera
h2	Parameter: heightTable
d	Parameter: distanceToTableCenter

Usage of Robotic Camera Arm for rotations around the x-axis

armLength	
Description	length of the robot arm
Syntax	float (in mm)
Default	
Context	FSI Viewer JS

Defines the length of the robot arm; default value is the distance of the camera to the camera target. Only relevant if hotspots are used.

armJointPosition	
Description	position of the arm joint
Syntax	float (in mm)
Default	
Context	FSI Viewer JS

Defines the position (x,y,z) of the robot arm joint; default value is equal to the camera targetPosition. Only relevant if hotspots are used.

armAngleMeasureSetup	
Description	angle where distance from camera to target is measured
Syntax	float (in degrees)
Default	0
Context	FSI Viewer JS

Defines the anglefrom which distance from camera to target was measured. Only relevant if hotspots are used.

armStartAngle	
Description	defines the start angle
Syntax	float (in degrees)
Default	-90
Context	FSI Viewer JS

Defines the start angle of the rotation around the x-axis. Only relevant if hotspots are used.

armEndAngle	
Description	defines the end angle
Syntax	float (in degrees)
Default	90
Context	FSI Viewer JS

Defines the end angle of the rotation around the x-axis. Only relevant if hotspots are used.

2.5.3 <virtualSpace> Section

The virtualSpace section is optional and only required if you want to verify the 3D projections for use with hot spots. Using the camera parameters, the viewer calculates the position of the hot spots based on the given 3D coordinates only.

cubeSizeX, cubeSizeY and cubeSizeZ	
Description	"cube" is the bounding box of the spin object in millimeters
Syntax	float
Default	-
Context	FSI Viewer JS

The "cube" is the bounding box of the spin object in millimeters.

cubeLineWidth	
Description	line width of the cube
Syntax	float
Default	-
Context	FSI Viewer JS

Line width of the drawn cube. Only applies if the parameter "drawCube" is activated.

baseRotationY, baseRotationX, baseRotationZ	
Description	specifies a rotation offset in degrees
Syntax	float
Default	0
Context	FSI Viewer JS

You can specify a rotation offset in degrees.
This is required only, if you started the 360° shooting session at an angle other than 0°.

rotationXRange, rotationYRange	
Description	specifies the range of the rotation
Syntax	float
Default	180 (X) or 360 (Y)
Context	FSI Viewer JS

Specifies the range of the rotation of a certain axis.

shiftX, shiftY and shiftZ	
Description	defines an offset of the spin object on the turn table in mm
Syntax	float
Default	-
Context	FSI Viewer JS

Optionally defines an offset of the spin object on the turn table in mm.

position	
Description	defines an offset of the spin object on the turn table in mm
Syntax	float
Default	-
Context	FSI Viewer JS

Alternative parameter to shiftX,shiftY, shiftZ - positon(x,y,z) is optional and defines an offset of the spin object on the turn table in mm.

2.5.4 <hotspots3d> Section

The hotspots3d section is optional and only required if you want to display hot spots on your 360° object spin. Hot Spots consist of a marker line and a marker dot as well as content area. The content area may contain all kind of HTML content, e.g. formatted text, hyperlinks, images, videos and alike.

Each hotspot requires an individual hot spot definition node.
In addition a <defaults> node can be used to define default parameter values.

HotSpot definition mode

```

<hotspots3d>
  <defaults>
    <normalClass value="hotspotNormal"/>
    <hoverClass value="hotspotOver"/>
    <activeClass value="hotspotAction"/>
    <perimeterAngleFadeOut value="15" />
  </defaults>
  <hotspot>
    <text value="Hot Spot A " />
    <x value="11"/>
    <y value="223"/>
    <z value="-116"/>
    <perimeterAngle value="0"/>
    <perimeterAngleRange value="60"/>
  </hotspot>
  <hotspot>
    <text value="Hot Spot B " />
    <x value="11"/>
    <y value="223"/>
    <z value="-116"/>
    <perimeterAngle value="90"/>
    <perimeterAngleRange value="30"/>
    <perimeterAngleFadeOut value="5" />
  </hotspot>
  [...]

```

Hot Spot Parameters:

x, y and z	
Description	x, y, z distance on the real object measured from center of object
Syntax	float (in mm)
Default	-
Context	FSI Viewer JS

The x, y and z distance in mm on the real object measured from the center of the object (usually equal to the center of the turn table). Combined with the required <camera> parameters, the viewer calculates the rotated and projected position of the hot spot on screen.

text	
Description	HTML content to display for this hot spot
Syntax	string
Default	""
Context	FSI Viewer JS

The HTML content to display for this hot spot.

Note that the value needs to be XML encoded if the content contains invalid characters for XML attribute values (e.g. '&' for '&', '<' for '<', '>' for '>' and '"' for '"').

horizontalMarkerSize	
Description	length of the horizontal line of the hot spot's marker
Syntax	integer
Default	30
Context	FSI Viewer JS

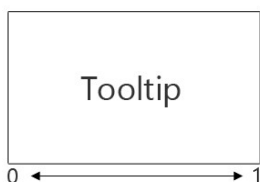
Length of the horizontal line of the hot spot's marker. Range: [-x ... 0 ... x]

verticalMarkerSize	
Description	length of the vertical line of the hot spot's marker
Syntax	integer
Default	100
Context	FSI Viewer JS

Length of the vertical line of the hot spot's marker. Range: [0 ... x]

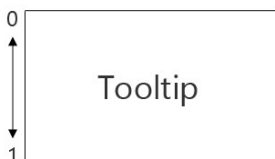
horizontalMarkerPosition	
Description	horizontal position of the marker line
Syntax	float
Default	0
Context	FSI Viewer JS

Defines the horizontal position of the marker line on the tooltip. Range: [0.0 ... 1.0] The value 0.0 marks the left corner of the tooltip while 1 marks the right corner.



verticalMarkerPosition	
Description	vertical position of the marker line
Syntax	float
Default	0.5
Context	FSI Viewer JS

Defines the vertical position of the marker line on the tooltip. Range: [0.0 ... 1.0] The value 0.0 marks the top corner of the tooltip while 1 marks the bottom corner.



markerAngle	
Description	angle of the vertical hot spot marker
Syntax	float
Default	45
Context	FSI Viewer JS

Defines the angle of the vertical hot spot marker.

perimeterAngle	
Description	viewing angle in degrees at which to display the hot spot
Syntax	float
Default	-
Context	FSI Viewer JS

The viewing angle in degrees at which to display the hot spot.
E.g. for a hot spot on the left side of the object use "90".
For a hot spot at the back of the object use "180".
Alias: perimeterAngleY

perimeterAngleX	
Description	viewing angle in degrees at which to display the hot spot
Syntax	float
Default	-
Context	FSI Viewer JS

The viewing angle in degrees at which to display the hot spot.
Only applies if a rotation around the x-axis is used.

perimeterAngleRange	
Description	angle range in which the hot spot shall be visible
Syntax	float
Default	30
Context	FSI Viewer JS

The angle range in which the hot spot shall be visible.
 E.g. a `perimeterAngle` of "90" and a `perimeterAngleRange` of "40" would display the hot spot from 70 to 110 degrees (+/- 20°).
 Alias: `perimeterAngleRangeY`

perimeterAngleRangeX	
Description	angle range in which the hot spot shall be visible
Syntax	float
Default	30
Context	FSI Viewer JS

The angle range in which the hot spot shall be visible.
 Only applies if a rotation around the x-axis is used.

perimeterAngleFadeOut	
Description	angle range in which the hot spot shall be faded in and out
Syntax	float
Default	10
Context	FSI Viewer JS

The angle range in which the hot spot shall be faded in and out.
 A value of "0" means that the hot spot will be displayed or hidden immediately when entering or leaving the `perimeterAngleRange`. Values greater than 0 extend the `perimeterAngleRange` by the given degrees in which the hot spot's opacity changes.
 Alias: `perimeterFadeOutY`

perimeterAngleFadeOutX	
Description	angle range in which the hot spot shall be faded in and out
Syntax	float
Default	10
Context	FSI Viewer JS

The angle range in which the hot spot shall be faded in and out. A value of "0" means that the hot spot will be displayed or hidden immediately when entering or leaving the perimeterAngleRange. Values greater than 0 extend the perimeterAngleRange by the given degrees in which the hot spot's opacity changes. Only applies if a rotation around the x-axis is used.

perimeterZoom	
Description	specifies from which magnification level on a hot spot is displayed
Syntax	float
Default	1.0
Context	FSI Viewer JS

An optional parameter specifying from which magnification level on a hot spot should be displayed. E.g. 2.5 would hide the hot spot as long as the magnification level is below 2.5. This parameters has no effect if zoom is disabled by the global "enableZoom" parameter.

perimeterImageWidth	
Description	defines the absolute width of spin instance for displaying hot spot
Syntax	float
Default	1.0
Context	FSI Viewer JS

Similar to perimeterZoom. While perimeterZoom depends on the magnification this parameter defines the absolute width of the (zoomed) spin instance to

decide whether to show or hide a tooltip. It is recommend to set this parameter as a default hot spot parameter when using relative instance sizes.

interactive	
Description	enables or disables reactions of hot spot areas to interactive actions
Syntax	boolean
Default	false
Context	FSI Viewer JS

The hot spot content area will not react on mouse over, click actions and alike when setting "interactive" to false.

actionParameter	
Description	value will be passed on to the hot spot callback function
Syntax	string
Default	""
Context	FSI Viewer JS

An optional parameter value that will be passed on to the hot spot callback function.

url	
Description	An optional URL to open if the user clicks the hot spot label.
Syntax	string
Default	""
Context	FSI Viewer JS

An optional URL to open if the user clicks the hot spot label. You can use any URL you can use with the window.open command, like regular URLs as well as "javascript:" or "email:" URLs.

urlTarget	
Description	target frame or window to open the URL in
Syntax	string
Default	_self
Context	FSI Viewer JS

The target frame or window to open the URL in.
Use e.g. "_blank" to open hyperlinks in a new window.

focusOnClick	
Description	zoom and/or rotate to hotspot position on click
Syntax	boolean
Default	true
Context	FSI Viewer JS

Enables zooming and/or rotating to hotspot position on click.

rotateOnFocus	
Description	rotate to sceneSet and scene
Syntax	boolean
Default	true
Context	FSI Viewer JS

Setting this parameter to true rotates to sceneSet and scene matching the coordinates defined in hotspot.perimeterAngle, while false keeps the current sceneSet and scene.

zoomOnFocus	
Description	zoom to hotspot
Syntax	boolean
Default	true
Context	FSI Viewer JS

Setting this parameter to true zooms to the hotspot, while false keeps the current zoom.

focusRadius	
Description	size of section when zooming to a hotspot
Syntax	float
Default	0.25
Context	FSI Viewer JS

Defines the size of the section when zooming to a hotspot (0...1).

focusAngleX, focusAngleY	
Description	defines scene with given focus angle (optional)
Syntax	float
Default	-
Context	FSI Viewer JS

If set, the target scene or sceneSet will not be determined by the perimeterAngle parameter, but with the given focusAngle.

normalClass	
Description	CSS class name to use for content <div> tag in normal state
Syntax	string
Default	"FSI360HotSpot_Normal"
Context	FSI Viewer JS

A CSS class name to use for the content <div> tag in normal state.

activeClass	
Description	CSS class name to use for content <div> tag in mouse onclick state
Syntax	string
Default	"FSI360HotSpot_Active"
Context	FSI Viewer JS

A CSS class name to use for the content <div> tag in onclick state.

hoverClass	
Description	CSS class name to use for content <div> tag in mouse over state
Syntax	string
Default	"FSI360HotSpot_Active"
Context	FSI Viewer JS

A CSS class name to use for the content <div> tag in mouse over state.

Example CSS classes:

It is recommended to add the following CSS classes to the HTML document containing FSI Viewer JS instances with hot spots. These classes will be used if you do not define "normalClass", "hoverClass" and "activeClass" parameters for your hot spots. The _Mobile und _SmallMobile classes will be added to the hot spot class names on mobile device or small mobile devices (phones). You can

use them to adjust the font size to increase the readability of text within the hot spots.

Example CSS classes

```
<style type="text/css">

div.FSI360HotSpot_Normal{
    border:2px solid rgba(255, 255, 0, 0.8);
    font-family:Arial,Helvetica,sans-serif;
    color:#000;
    font-size:13px;
    background: rgba(255, 255, 200, 0.8);
    padding:3px;
    border-radius: 3px;
    cursor:pointer;
    box-shadow: 4px 4px 4px rgba(0,0,0,0.3);
    white-space:nowrap;
}

div.FSI360HotSpot_Active{
    background: #FFF;
}

div.FSI360HotSpot_Mobile{
    font-size:13px;
}

div.FSI360HotSpot_SmallMobile{
    font-size:32px;
}

</style>
```

cssClass	
Description	CSS class name to add to the hot spot's content <div> tag
Syntax	string
Default	""
Context	FSI Viewer JS

An additional CSS class name to add to the hot spot's content <div> tag.

cssStyle	
Description	CSS style value to add to the hot spot's content <div> tag
Syntax	string
Default	""
Context	FSI Viewer JS

A additional CSS style value to add to the hot spot's content <div> tag.

2.5.5 <video> Section

The video section is optional and only required if you want to enable ultra smooth rotations by providing a video of the spin.

source	
Description	CSS style value to add to the hot spot's content <div> tag
Syntax	string
Default	""
Context	FSI Viewer JS

The value may be an absolute URL. Otherwise it will be relative to the [viewer] /applications/spin360/videos/ folder.

NOTE: Just provide the file name WITHOUT the file extension.

Please refer to "Creating Videos" below for more information on spin videos.

sceneSet	
Description	defines in which Scene Set the video is used
Syntax	string
Default	""
Context	FSI Viewer JS

If Scene Sets are used, this parameter defines in which Scene Set the video will be used for smooth rotation.

base	
Description	optional prefix for all "source" parameters
Syntax	string
Default	""
Context	FSI Viewer JS

An optional prefix for all "source" parameters.

spinDirection	
Description	defines the spin direction of the video
Syntax	String "left" or "right"
Default	"left"
Context	FSI Viewer JS

Defines the spin direction of the video.
You might want to change the spinDirection option if the object spins into the wrong direction when dragging the mouse left or right.

captureFrames	
Description	number of frames to capture from the video
Syntax	integer
Default	120
Context	FSI Viewer JS

The number of frames to capture from the video. The more frames you define the smoother the rotation will appear. More frames require more time to capture the videos on startup. 120 frames (default) means 3° per frame.

captureOffsetStart	
Description	offset at start of video to use when capturing the video frames
Syntax	float (in seconds)
Default	0
Context	FSI Viewer JS

An optional offset at the start of the video to use when capturing the video frames. You might want to define an offset if the video does not start exactly at the position of the first image (0°).

captureOffsetEnd	
Description	offset at end of video to use when capturing the video frames
Syntax	float (in seconds)
Default	0
Context	FSI Viewer JS

An optional offset at the end of the video to use when capturing the video frames. You might want to define an offset if the video does not start exactly at the position of the first image (0°).

widthCorrectionFactor	
Description	adjusts width of the video to width of the images on screen
Syntax	float
Default	1.0
Context	FSI Viewer JS

An optional factor to adjust the width of the video to the width of the images on screen. You might want to adjust the value if the image do not match the video dimension.

heightCorrectionFactor	
Description	adjusts height of the video to height of the images on screen
Syntax	float
Default	1.0
Context	FSI Viewer JS

An optional factor to adjust the height of the video to the height of the images on screen. You might want to adjust the value if the image do not match the video dimension.

xCorrectionFactor	
Description	adjusts horizontal position of video to horizontal position of images
Syntax	float
Default	0.0
Context	FSI Viewer JS

An optional factor to adjust the horizontal position of the video to the horizontal position of the images on screen. You might want to adjust the value if the image do not match the video position.

yCorrectionFactor	
Description	adjusts vertical position of video to vertical position of images
Syntax	float
Default	0.0
Context	FSI Viewer JS

An optional factor to adjust the vertical position of the video to the vertical position of the images on screen. You might want to adjust the value if the image do not match the video position.

frameOffset	
Description	determines a frame offset if required
Syntax	integer
Default	0
Context	FSI Viewer JS

Determines a frame offset at the start if required.

2.5.6 Plugin Parameters

plugins	
Description	adds a Plug-in (e.g. Fullscreen) to the viewer
Syntax	String
Default	""
Context	FSI Viewer JS

Adds a plug-in to the viewer.

Please note:

If you would like to add specific parameters to the plugins via the fsi-viewer tag, you can do this by prefixing the parameter name with the name of the plug-in, e.g. `autospin_enabled="true"`

FullScreen Plug-In

Adding the parameter `plugins: "FullScreen, Resize"` enables a full screen view option in FSI Viewer JS. The `Resize` parameter needs to be set in order to ensure proper adjusting to the full screen.

NOTE: If the FSI Viewer JS is embedded in an `iFrame`, the attribute `allowfullscreen="true"` needs to be added to the `iFrame`. Otherwise the FullScreen Plug-in will not work due to security restrictions of the browser.

AutoSpin Plug-In

available from version 16.09.30

Adding the parameter `plugins: "AutoSpin"` enables a button which lets you play/pause the animation that is set with the `autospinspeed` parameter. The following parameters can be set:

Speed	
Description	Time in seconds for a full rotation
Syntax	float
Default	-
Context	FSI Viewer JS

Has the same function as `→ FSI Viewer JS Parameters`, but is used here in the context of the plug-in. Time in seconds for a full rotation. Negative values reverse spin direction.

preventClickZoomWhileSpinning	
Description	prevents zooming while Autospin is activated
Syntax	Boolean
Default	false
Context	FSI Viewer JS

Prevents zooming while clicking into the Autospin.

Button	
Description	enables Play/Pause button
Syntax	Boolean
Default	true
Context	FSI Viewer JS

Enables the Play/ Pause button for the autospin animation

MouseModes Plug-In

The MouseMode plugin is activated by default with the `_default.xml`.





The following parameters can be set:

MenuOffset	
Description	Indentation of the menu buttons
Syntax	Integer
Default	0
Context	FSI Viewer JS

Specifies the space in pixels left of the menu button(s) of the plug-in.

Mode <i>n</i>	
Description	Removes the button for mouse mode <i>n</i> from the menu bar
Syntax	Boolean
Default	-
Context	FSI Viewer JS

Provides the possibility to hide specific mouse mode buttons.

Mode n		Mouse Mode
Mode0		Zoom
Mode1		Pan
Mode2		Rotate 3D (X/Y Axis)
Mode3		Rotate 2D (Z Axis)

Sequence	
Description	Sequence of the buttons
Syntax	String
Default	0,1,3,2
Context	FSI Viewer JS

You can alter the sequence of the buttons by providing the modes separated by commas (see the table above).

Example: "1,0,2,3" alters the sequence of the buttons to "Pan, Zoom, Rotate 3D, Rotate 2D".

2.6 Creating and using Spin Videos

When creating spin videos for ultra smooth rotations it is strongly recommended to use a camera that can take photos and record videos so that the position and technical specifications apply to both, the video and the photos.

Make sure you do not modify the object or camera position between the video and photo shooting session.

- 1) Take the photos from different viewing angles
- 2) Record a video of a full, continuous object spin starting at the angle you took the first photo from until reaching the same angle again. You might want to take a video of 2 full spins and cut the video later on.
- 3) Cut the video if required making sure the video starts and ends at the viewing angle you took the first photo at
- 4) Export the video in the following formats:
webm (for Chrome, Firefox, Safari, etc.)
mp4 (for MS IE)
- 5) Copy the video files to either a static source connector on FSI Server or to the [viewer]/applications/spin360/videos/ folder using the same file names just with the file extension matching the video format
- 6) Add the <video> section to your configuration file and add the "source" parameter (just the file name WITHOUT the file extension).

2.7 Embedding the viewer at runtime via JavaScript

It is also possible to embed the viewer at runtime via JavaScript.

I) Passing parameters via node attribute:

Example:

```
var node = document.createElement("fsi-viewer");

node.setAttribute("width", "400");
node.setAttribute("height", "600");
node.setAttribute("src", "images/Rocker Shoe.tif");
node.setAttribute("debug", "true");

document.body.appendChild(node);

$FSI.initCustomNode(node);
// Important: call this AFTER adding the node to the DOM
```

I) Passing parameters via JavaScript object:

NOTE: In this case any parameters passed via node attribute will be ignored.

Example:

```
var node = document.createElement("fsi-viewer");
node.style.width = "600px";
node.style.height = "400px";

var parameters =
{"imagesrc" : "images/Rocker Shoe.tif", "skin":"silver",
"debug":true};

$FSI.setParameters(node, parameters);
document.body.appendChild(node);

// Important: call this AFTER adding the node to the DOM
$FSI.initCustomNode(node);
```

Note:

`$FSI.initCustomNode(node)` initializes the given node.

When adding multiple viewers you can initialize all new FSI Viewer JS nodes in one go, using this command:

```
$FSI.initCustomTag("fsi-viewer");
```

or, when using different custom tags (fsi-viewer, fsi-imageflow,...):

```
$FSI.initCustomTags();
```

 to initialize all new FSI custom tags.**Destroying Instances**

Before finally removing `<fsi-viewer>` nodes from the DOM, you need to call the `destroy()` method of the node.

This will release all event handlers and free allocated resources.

Example:

```
var node = document.getElementById("myViewer");

node.destroy();

node.parentNode.removeChild(node);
```


2.8 JavaScript Interface

The JavaScript interface of FSI Viewer JS can be useful if you want to change the current image or config or add additional content depending on the object spin position or hot spot actions.

2.8.1 Public Methods

void	changeConfig(strCfgFileName, oParameters);
	Reset the viewer object and change the configuration to the given configuration file. strCfgFileName (optional): path to the configuration file (see "cfg" parameter) oParameters (optional): an object containing parameters If present, these parameters will overwrite the parameters defined upon initialization.

Example:

```
var oParameters = {};  
oParameters.src = "images/bluemarble.tif";  
/*  
// optional:  
oParameters.imageWidth;  
oParameters.imageHeight;  
oParameters.cropRect;  
oParameters.cropValues;  
oParameters.effects;  
*/  
  
fsiviewer.changeConfig(oParameters);
```

bSuccess	changeImage(cfg, strView);
	Resets the viewer object and changes the image displayed to the given image file.

void	start();
Starts the viewer.	

void	printAPI();
Shows all API methods in the console in alphabetical order.	

void	init(elementID, oParameters, bDebug);
Initialises the chosen viewer element.	

bInitDone	getInitDone();
Returns if the initialization process is done.	

void	spinToDegree(fDegree);
Spins to the position indicated by degree. Note that you can only spin to a position that has a corresponding HD image, not to a position that shows a video only. <i>fDegree</i> : the desired position in degree (0 .. 360)	

void	spinToTarget(nTargetFrame, nStopSceneSet, nFrames);
Spins to the position indicated by the target parameters. Note that you can only spin to a position that has a corresponding HD image, not to a position that shows a video only.	

nMousemode	getMouseMode();
Returns the current active mouse mode.	

void	setMouseMode(nMode);
Sets the active mouse mode.	

nInstanceID	getInstanceID();
Returns the number of the viewer object.	

void	destroy();
Destroys the given FSI Viewer JS object. You should destroy the instance before you remove the object spins <div> tag from the DOM tree.	

strVersion	getVersion();
Returns a string containing the FSI Viewer JS software version.	

strVersion	getBuild();
Returns a string containing the FSI Viewer JS software build number.	

nHeight	getMenuHeight(strPurpose);
Returns a string containing the height of the menu bar.	

bSuccess	addListener(strListenerName, fn, iScope);
Calls a specific listener..	

bSuccess	removeListener(strListenerName, fn);
Removes a specific listener..	

void	showAboutWindow();
Shows the "About" window.	

void	showMenuButton(strButtonID, bShow);
Shows a specific menu button.	

void	updateSize();
Updates the size of a fsi-viewer element.	

void	resetView();
Reset the viewer to the initial magnification and rotation.	

fZoomInPercent	getZoom();
Returns the current magnification level..	

void	setZoom(fPercent, bPreliminary, bAnimate);
Set magnification to the amount indicated by fPercent [0...100], where fPercent = 0 means normal scale and fPercent = 100 for maximum magnification. bPreliminary: Set to true if the magnification level is temporary only. In this case no additional image data will be loaded. bAnimate: Set to true to animate from the current to the given magnification level.	

strViewString	getViewString();
	Returns the string representing the current image section and rotation. The format is: SceneSet, Scene, left, top, right, bottom. <i>SceneSet</i> is 1 for rotations around one axis. <i>Scene</i> describes the rotation. If you use 36 images for a spin, scene 18 means a rotation of 180 degree. <i>left, top, right, bottom</i> : image coordinates in the range [0.0 to 1.0].

bSuccess	gotoViewString(strViewString, bAnimated);
	Display the image section and rotation as specified by "strViewString". Please refer to the description of the "getViewString" method for a format description. The boolean "bAnimated" can be used to specify, whether the view should be displayed immediately or in form of an animated movement from the current view to the specified view.

strViewString	getHotspotViewString(nHotspotID)
	returns the view string for the given hotspot. FSIHotspotID is the consecutive number of the Hotspot (1,2,3,..)

bSuccess	gotoHotspot(FSIHotspotID, bAnimate)
	displays the image section as specified by the Hotspot. FSIHotspotID is the consecutive number of the Hotspot (1,2,3,..) The boolean "bAnimated" can be used to specify, whether the view should be displayed immediately or in form of an animated movement from the current view to the specified view.

bPresent	getMenuButtonPresent(strButtonID)
	Shows active menu buttons, can be used to remove inactive buttons from the menu if you are using a custom skin. NOTE: Method only delivers correct results after the callback "onInitMenu" is called.

void	pressMenuButton(strButtonID)
	Sets the menu button to active.

void	clickMenuButton(strButtonID)
	Sets the menu button to active.

void	releaseMenuButton(strButtonID)
	Disables the active menu button.

void	registerExternalMenuButton(elButton, strButtonID, strTip)
	Registers an external menu button.

void	unregisterExternalMenuButton(elButton)
	Unregisters an external menu button.

void	registerExternalMenuButtonsFromContainer(elContainer)
	Registers an external menu button from a specific container.

void	unregisterExternalMenuButtonsFromContainer (elContainer)
	Unregisters an external menu button from a specific container.

strClassName	getSkinClassName()
	Returns the name of the current used skin class, e.g. fsi-skin-black

bFullscreen	getInFullScreenMode()
	Enables the FullScreen mode.

bSuccess	setFullScreenElement(elFullScreenContainer)
	Defines the FullScreen element.

nFrames	getSpinFrames()
	Returns the amount of spin frames.

bTransparent	getImageIsTransparent()
	Returns the info if the image is transparent.

oDimension	getSrcDim()
	Returns the dimensions of the source image.

void	showMenuToolTip(strTip, elSrc, x, y)
<p>Shows custom tooltips.. strTip: HTML code of the tooltip or ID in language.xml files elSrc: the DOM element which is used to align the tooltip, usually event.target x,y: (optional) offset for tooltip position</p>	

void	hideMenuToolTip()
<p>Shows custom tooltips.. strTip: HTML code of the tooltip or ID in language.xml files elSrc: the DOM element which is used to align the tooltip, usually event.target x,y: (optional) offset for tooltip position</p>	

oParameters	getParameters();
<p>Returns an object containing all parameters set via javascript or custom tag attributes.</p>	

mixed	getConfigValue(strName);
<p>Returns the value of the configuration parameter "strName".</p>	

bResult	makeBoolean(value, bDefaultValue);
<p>Transforms a value into a boolean value, e.g. "1" to true, "false" to false <i>bDefaultValue</i> is optional and false if not defined.</p>	

mixed	getPluginParameter(strPluginName, strParameterName, defaultValue, bMakeBoolean);
<p>Returns the parameter defined for the given plugin. <i>defaultValue</i> and <i>bMakeBoolean</i> are optional. example: <code>iViewer.getPluginParameter("pages", "dir");</code></p>	

2.8.2 Callbacks

You can define callback functions by assigning methods to the FSI Viewer JS instance.

Using Callbacks

Method A – Create a function in window scope

Example:

```
<script>
function onFSIViewerViewChanged(strView) {
  console.log(strView);
}
</script>
```

a) assignment in Custom Tag:

Example:

```
<fsi-viewer onViewChanged="onFSIViewerViewChanged" [...] />
```

or

b) assignment in XML configuration file:

Example:

```
<fsi_parameter>
<onViewChanged value="onFSIViewerViewChanged" />
[...]
</fsi_parameter>
```

Method B - Assignment by parameter

Example:

```
oParameters= {};
oParameters.onViewChanged = function(strView){
  console.log(strView);
}

var myViewer = new $FSI.Viewer(elContainer, oParameters);
myViewer.start();
// all callback assignments must take place before starting
```

Method C - Assignment by instance property**Example:**

```
var myViewer = new $FSI.Viewer(elContainer, oParameters);
myViewer.onViewChanged = function(strView){
  console.log(strView);
}

myViewer.start();
// all callback assignments must take place before starting
```



Please note: You can modify or add new callbacks when calling `changeConfig()`. You cannot change, add or remove callbacks after the viewer started.

The following callback methods exist:

	onReady();
	Called as soon as FSI Viewer finished loading data and gets interactive.

	onDestroy();
	Called when the fsi-viewer element is destroyed.

onMenuEnabled(bEnabled);
Called if the menu is enabled/disabled.

onMenuReady();
Called as soon as the menu is ready.

onMenuDestroy();
Called if the menu is destroyed.

onZoomChanged(fScale, fScaleMax);
This optional callback function will be called each time the magnification level changes. <i>fScale</i> is the magnification level from 0 to 100. <i>fScaleMax</i> sets the maximum zoom level.

onAnimationComplete();
Called when a spin and/or zoom animation finishes.

onAnimationStart();
Called when a spin and/or zoom animation starts.

onChangeConfig(strCfgFileName, oParameters);
Called when the config file is changed.

onChangeImage(oParameters, strView);
Called when the image is changed.

	onResize(nWidth, nHeight, bResizeDone, BFullScreen);
Called when the fsi-viewer element is resized.	
	onSkinChanged(strCurrentSkinClass, strCurrentSkinClassBefore)
This optional callback function can be used to call an action as soon as the skin class is changed.	
	onRegisterExternalMenuButtons()
This optional callback function can be used to call an action as soon external menu buttons are registered.	
	onAfterRegisterExternalButton(elButton, strButtonID, bButtonPresentInViewer)
This optional callback function can be used to remove inactive buttons from the menu bar if you are using a custom skin.	
	onProgress(fPercent);
This optional callback function can be called while the loading progress.	
	onMenuButtonEnabled(strButtonID, null, bEnable);
This optional callback function can be used call an action when a button is enabled or disabled.	
	onMenuButtonPressed(strButtonID, evt);
This optional callback function can be used to set an action that starts when a button is pressed.	

onMenuButtonReleased(strButtonID, evt);

This optional callback function can be used to set an action when a certain button is released..

onMenuButtonClicked(strButtonID, evt);

This optional callback function can be used to set an action when a certain button is released..

onMenuButtonSetPressed(bPressed);

This optional callback function can be used to call an action when a button is pressed.

onReset();

This optional callback function can be used to call an action when the configuration is changed.

onInitMenu();

This optional callback function can be used to set an action when the menu is built (after all buttons are added).

onMouseModeChanged(nMouseMode);

This optional callback function can be used to call an action when the mouse mode is changed.

onStartDragging(bInNavWindow);

This optional callback function can be used to call an action when the user starts dragging the image.

onStopDragging(bInNavWindow);
This optional callback function can be used to call an action when the user stops dragging the image.

onDataComplete();
This optional callback function can be used to call an action while the configuration data loading is completed. Images/videos are not loaded yet at this point.

onFullScreen(bFullScreenActive);
This optional callback function can be used to call an action when FullScreen mode is enabled or disabled.

onViewChanged(strViewString);
This optional callback function can be used to call an action as soon as the viewed image section changes.

onHotspotEvent(strAction, nHotspotID, strHotspotAttributeID, strContent)
This optional callback function can be used to call an action as soon as a certain Hotspot event occurs. The following events are possible: mouseover, mousedown, mouseout & click.

3 FSI TouchZoom

FSI TouchZoom is an HTML5 and Javascript based tool adding image zoom to touch enabled devices like Apple iPad™. When pinch zooming into a website containing single source images provided by FSI Server, the images will be displayed in the resolution matching the current magnification.

Supported devices

- iOS 4 or higher
- Android 4.x based devices
- Windows 8 with touch enabled browsers
- Mac OS with touch device

Usage

To use FSI TouchZoom, make sure the respective images on your website come from FSI Server and add the following script to the <head> section of your website:

Example:

```
<head>
[...]
<script src="/viewer/applications/touchzoom/js/fsitouchzoom.js"
type="text/javascript">
</script>
</head>
```

It is possible to pass optional parameters via JavaScript. In this case it is crucial to embed the TouchZoom script before defining the parameters:

Example:

```
<script type="text/javascript"
src="/viewer/applications/touchzoom/js/fsitouchzoom.js"></script>

<script type="text/javascript">
$FSI.touchZoomParameters =
{debug:true, useDevicePixelRatio:false};
</script>
```


Alternatively it is possible to use an external JavaScript for defining the parameters. This is useful if the parameters are valid for multiple HTML documents.

In case the script is automatically initialised (default), the automatically created instance can be addressed via

```
$FSI.touchZoom
```

3.1 Initializing FSI TouchZoom manually

Initializing or adding images manually

If you are adding images by script, you can decide to:

I) let `FSI.TouchZoom` initialize automatically and add image manually:

Example:

```
<script type="text/javascript
src="/viewer/applications/touchzoom/js/fsitouchzoom.js"></script>

<script type="text/javascript">

addMyImagesToTheDom(); // add your images here

// make FSI.TouchZoom look for new images

$FSI.touchZoom.scanForNewImages();

</script>
```

II) initialize `FSI.TouchZoom` manually after adding your images:

Example:

```
<script type="text/javascript"
src="/viewer/applications/touchzoom/js/fsitouchzoom.js"></script>

<script type="text/javascript">

// prevent $FSI.TouchZoom from initializing automatically
// on document load
$FSI.touchZoomParameters = {autoInit:false};

function onBodyLoaded()
{
addMyImagesToTheDom(); // add your images here

// create and initialize FSI.TouchZoom
var parameters = {debug:true, useDevicePixelRatio:true};

myTouchZoom = new $FSI.TouchZoom();
myTouchZoom.init(parameters);

}

</script>
```

3.2 FSI TouchZoom Parameters

You can optionally modify the way FSI TouchZoom works by passing parameters to the script. To do so, pass an object containing the parameters and value as an object to the constructor:

Example:

```
<script type="text/javascript">
FSITouchZoom = new $FSI.TouchZoom({debug:true,
zoomPrecision:6});
FSITouchZoom.init();
</script>
```

debug	
Description	displays debug & status information in the javascript console.
Syntax	Boolean
Default	false
Context	FSI TouchZoom

Display debug and status information in the browser's javascript console.

useDevicePixelRatio	
Description	Use the device's pixel ratio to display even sharper images.
Syntax	Boolean
Default	true
Context	FSI TouchZoom

Use the device's pixel ratio to display even sharper images. For devices with a pixel ratio > 1 (e.g. Apple iPad 3) the images will be displayed in higher resolutions. If the aspect ratio is 2, the images loaded when pinching a page will be 2 x the resolution compared to devices with a pixel aspect ratio of 1.

monitorPositions	
Description	The script monitors the position of all single source tags
Syntax	Boolean
Default	true
Context	FSI TouchZoom

By default the script monitors the position of all single source tags to ensure that a higher resolution image will be loaded if the image is within the viewport. For huge amounts of images on a single web page you might want to disable this feature, if the positions of the images do not change (e.g. there is no javascript modifying the DOM tree).

useTiledImages	
Description	Zoomed images will be tiled into multiple image tiles
Syntax	Boolean
Default	true
Context	FSI TouchZoom

By default, zoomed images of 1000 pixel in width or height will be tiled into multiple image tiles. While this increases the user experience and the loading performance, you might want to disable this feature if you e.g. change the style of an image (e.g. `style.display`) at runtime. In this case the change would not effect the tiled image.

zoomPrecision	
Description	This parameter defines the precision of loading zoomed images.
Syntax	Integer
Default	4
Context	FSI TouchZoom

This parameter defines the precision of loading zoomed images. Using the value "0" will load images exactly matching the current magnification. This does on the other hand prevent effective caching. Values greater than 0 increase the precision how often the script loads new images when changing the magnification. The value 4 means that the script loads new images at 1.0, 1.25, 1.5, 1.75, 2.0 etc. magnification levels.

loadingImage	
Description	Specifies which image to display at the bottom right of the viewport
Syntax	Boolean or URL
Default	true
Context	FSI TouchZoom

Specifies whether or which image to display at the bottom right of the viewport while new images are being retrieved from FSI Server. You can use true or false to enable/disable the loading image display or provide an URL to a custom image.

enableByCSSClass	
Description	provides a comma separated list of CSS class names
Syntax	String
Default	""
Context	FSI TouchZoom

Using the `enableByCSSClass` parameter you can provide a comma separated list of CSS class names. Only images containing at least one of the given class names will be modified by this script. All other images normal images that do not increase the resolution dynamically.

You can combine this parameter with the `disableByCSSClass` parameter.

disableByCSSClass	
Description	provides a comma separated list of CSS class names
Syntax	String
Default	""
Context	FSI TouchZoom

Using the `disableByCSSClass` parameter you can provide a comma separated list of CSS class names. Images containing at least one of the given class names will NOT be modified by this script. All other images containing single source image URLs will be modified.

You can combine this parameter with the `enableByCSSClass` parameter.

3.3 Initializing FSI TouchZoom manually

Initializing or adding images manually

If you are adding images by script, you can decide to:

I) let FSI.TouchZoom initialize automatically and add image manually:

Example:

```
<script type="text/javascript"
src="/viewer/applications/touchzoom/js/fsitouchzoom.js"></script>

<script type="text/javascript">

addMyImagesToTheDom(); // add your images here

// make FSI.TouchZoom look for new images

$FSI.touchZoom.scanForNewImages();

</script>
```

II) initialize FSI.TouchZoom manually after adding your images:**Example:**

```
<script type="text/javascript"
src="/viewer/applications/touchzoom/js/fsitouchzoom.js"></script>

<script type="text/javascript">

// prevent $FSI.TouchZoom from initializing automatically
// on document load
$FSI.touchZoomParameters = {autoInit:false};

function onBodyLoaded()

{

addMyImagesToTheDom(); // add your images here

// create and initialize FSI.TouchZoom
var parameters = {debug:true, useDevicePixelRatio:true};

myTouchZoom = new $FSI.TouchZoom();
myTouchZoom.init(parameters);

}

</script>
```

3.4 JavaScript Interface

Usually you will not need to use the javascript interface.

Even if you modify an images src parameter at runtime, the tool will notice that and replace the magnified image accordingly.

If you on the other hand add images to the DOM tree at runtime or modify the position of images by script, the following interface is available to update the images and positions:

```
FSITouchZoom = new $FSI.TouchZoom();  
FSITouchZoom.init();
```

I) Public Methods

void	scanForNewImages();
	Calling the scanForNewImages method will make the script look for single source images that have not been present in the DOM tree when the document finished loading. If you e.g. add tags at runtime, you need to call this method in order to make the script aware of new image(s).

void	addImage(elImg, strSrc);
	This adds a single node to the list of dynamic touch zoom images. elImg : the node object strSrc (optional): the image URL (img.src if undefined) Note: you can as well call <code>scanForNewImages()</code> after adding an node.

void	removeImages(arImageDOMElements, bSilent);
	Removes one or more touch zoom images and restores their original state. arImageDOMElements : an node object or an array of node objects

void	updateImagePositions();
This makes the tool update all image positions and refresh images if needed. You might want to call this after modifying the DOM tree by script.	

void	printAPI();
Shows all API methods in the console in alphabetical order.	

void	setZoomPrecision(fPrecision);
Use this method to modify the parameter zoomPrecision at runtime.	

void	destroy();
Destroys FSI TouchZoom.	

void	init();
Initializes FSI TouchZoom.	

strVersion	getVersion();
Returns a string containing the FSI ImageFlow software version.	

strVersion	getBuild();
Returns a string containing the FSI ImageFlow software build number.	

void	enable(bEnable);
Enables FSI TouchZoom.	

void	lockImage(eImage, bLock);
Locks a certain image .	

void	lockImageSourceImage(eImage, bLock);
Locks a certain image to it's source dimensions.	

void	lockUpdates(bLock);
Locks updates .	

void	resetImage(eImg);
Resets a certain image.	

void	triggerRefresh();
Triggers a refresh.	

4 FSI QuickZoom

FSI QuickZoom is an HTML5 and Javascript based tool adding image zoom to desktop and laptop computers on mouseover. When hovering over a single source image provided by FSI Server, the part of the image the cursor is currently pointing at will be displayed enlarged in a zoom window next to the original image.

Supported devices

- Desktop or laptop computers & all major browsers

Usage

To use FSI QuickZoom, make sure the respective images on your website come from FSI Server and add the following script to the <head> section of your website:

Example:

```
<head>
[...]
<script src="viewer/applications/quickzoom/js/fsiquickzoom.js"
type="text/javascript">
</script>
</head>
```

It is possible to pass optional parameters via JavaScript. In this case it is crucial to embed the QuickZoom script before defining the parameters:

Example:

```
<script type="text/javascript
src="/viewer/applications/quickzoom/js/fsiquickzoom.js"></script>

<script type="text/javascript">
$FSI.quickZoomParameters =
{debug:true, useDevicePixelRatio:false};
</script>
```

Alternatively it is possible to use an external JavaScript for defining the parameters. This is useful if the parameters are valid for multiple HTML documents.

In case the script is automatically initialised (default), the automatically created instance can be addressed via `$FSI.quickZoom`

4.1 How it works

FSI QuickZoom searches for all `` tags in the HTML document containing FSI Server single source image URLs and adds event handlers to them. When moving the mouse cursor over these images a magnified image will be displayed next to the original image. The user can move the cursor across the original image to display any image section of interest.

4.2 FSI QuickZoom Parameters

You can optionally modify the way FSI QuickZoom works by passing parameters to the script. To do so, pass an object containing the parameters and value as an object to the constructor:

Example:

```
<script type="text/javascript">
FSIQuickZoom = new $FSI.QuickZoom({debug:true,
magnification:6});
FSIQuickZoom.init();
</script>
```

debug	
Description	displays debug & status information in the javascript console.
Syntax	Boolean
Default	false
Context	FSI QuickZoom

Display debug and status information in the browser's javascript console.

useDevicePixelRatio	
Description	Use the device's pixel ratio to display even sharper images.
Syntax	Boolean
Default	true
Context	FSI QuickZoom

Use the device's pixel ratio to display even sharper images. For devices with a pixel ratio > 1 (e.g. Apple iPad 3) the images will be displayed in higher resolutions. If the aspect ratio is 2, the images loaded when pinching a page will be 2 x the resolution compared to devices with a pixel aspect ratio of 1.

headers	
Description	display debug and status information in the JavaScript console
Syntax	String
Default	"cors"
Context	FSI QuickZoom

The "&headers=" parameter to use for image requests.

enableByCSSClass	
Description	provides a comma separated list of CSS class names
Syntax	String
Default	""
Context	FSI QuickZoom

Using the enableByCSSClass parameter you can provide a comma separated list of CSS class names. Only images containing at least one of the given class names will be modified by this script. All other images are displayed as normal images that do not use FSI QuickZoom.

You can combine this parameter with the disableByCSSClass parameter.

disableByCSSClass	
Description	provides a comma separated list of CSS class names
Syntax	String
Default	""
Context	FSI QuickZoom

Using the `disableByCSSClass` parameter you can provide a comma separated list of CSS class names. Images containing at least one of the given class names will NOT be modified by this script. All other images containing single source image URLs will be modified.

You can combine this parameter with the `enableByCSSClass` parameter.

addCSSClass	
Description	adds an additional CSS class
Syntax	String
Default	""
Context	FSI QuickZoom

Using the `addCSSClass` parameter you can add an additional CSS class. Only images containing the given class names will be modified by this script. All other images are displayed as normal images that do not use FSI QuickZoom.

You can combine this parameter with the `disableByCSSClass` parameter.

magnification	
Description	defines the magnification level
Syntax	Float
Default	""
Context	FSI QuickZoom

Defines the magnification level of the image section displayed in the zoom window.

It is also possible to define the magnification for each image tag individually by adding `FSIQuickZoomMagnification`.

Example:

```
<img FSIQuickZoomMagnification="4" src= [...]>
```

maxZoomWindowWidth	
Description	defines the maximum zoom window width
Syntax	Float
Default	-
Context	FSI QuickZoom

Defines the maximum width of the zoom window displayed next to the original image.

maxZoomWindowHeight	
Description	defines the maximum zoom window height
Syntax	Float
Default	-
Context	FSI QuickZoom

Defines the maximum height of the zoom window displayed next to the original image.

minZoomWindowWidth	
Description	defines the minimum zoom window width
Syntax	Float
Default	-
Context	FSI QuickZoom

Defines the minimum width of the zoom window displayed next to the original image.

minZoomWindowHeight	
Description	defines the minimum zoom window height
Syntax	Float
Default	-
Context	FSI QuickZoom

Defines the minimum height of the zoom window displayed next to the original image.

skipImageWidth	
Description	defines minimum image width for QuickZoom use
Syntax	Boolean
Default	-
Context	FSI QuickZoom

If the image width is smaller than the width defined with this parameter, FSI QuickZoom will not be applied to it.

skipImageHeight	
Description	defines minimum image height for QuickZoom use
Syntax	Boolean
Default	-
Context	FSI QuickZoom

If the image height is smaller than the height defined with this parameter, FSI QuickZoom will not be applied to it.

pageMargin	
Description	offset for the zoom window to each side of the page
Syntax	Float
Default	-
Context	FSI QuickZoom

Defines the offset for the zoom window to each side of the page.

zoomImageBorderWidth	
Description	defines the image border width
Syntax	Float
Default	-1
Context	FSI QuickZoom

Defines the image border width of the zoom window.

zoomImageOffset	
Description	defines the offset of the zoom window to the original image
Syntax	Float
Default	-1
Context	FSI QuickZoom

Defines the horizontal offset of the zoom window to the original image.

Defining box-shadow and fade-in

The box-shadow and the fade-in of the FSI QuickZoom zoom window can be defined via CSS.

Example:

```
div.FSIQuickZoom {
border:1px solid #666;
background-color: white;
box-shadow: 3px 3px 3px rgba(0,0,0,0.25);
animation-name: QuickZoomFadeIn;
animation-duration: 0.5s;
}

@keyframes QuickZoomFadeIn{
from {
opacity: 0;
}
to {
opacity: 1;
}
}
```

4.3 Initializing FSI QuickZoom manually

Initializing or adding images manually

If you are adding images by script, you can decide to:

I) let FSI.QuickZoom initialize automatically and add image manually:

Example:

```
<script type="text/javascript
src="/viewer/applications/quickzoom/js/fsiquickzoom.js"></script>

<script type="text/javascript">

addMyImagesToTheDom(); // add your images here

// make FSI.QuickZoom look for new images

$FSI.quickZoom.scanForNewImages();

</script>
```

II) initialize FSI.QuickZoom manually after adding your images:

Example:

```
<script type="text/javascript"
src="/viewer/applications/quickzoom/js/fsiquickzoom.js"></script>

<script type="text/javascript">

// prevent $FSI.QuickZoom from initializing automatically
// on document load
$FSI.quickZoomParameters = {autoInit:false};

function onBodyLoaded()
{
addMyImagesToTheDom(); // add your images here

// create and initialize FSI.QuickZoom
var parameters = {debug:true, useDevicePixelRatio:true};

myQuickZoom = new $FSI.QuickZoom();
myQuickZoom.init(parameters);
}

</script>
```

4.4 JavaScript Interface

Usually you will not need to use the javascript interface.

Even if you modify an images src parameter at runtime, the tool will notice that and replace the magnified image accordingly.

If you on the other hand add images to the DOM tree at runtime or modify the position of images by script, the following interface is available to update the images and positions:

```
FSIQuickZoom = new $FSI.QuickZoom();
FSIQuickZoom.init();
```

I) Public Methods

void	init();
Initializes FSI QuickZoom.	

void	printAPI();
Shows all API methods in the console in alphabetical order.	

void	destroy();
Destroys FSI QuickZoom.	

void	addImages(elImg, strSrc);
This adds a single <code></code> node to the list of dynamic touch zoom images. <code>elImg</code> : the <code></code> node object <code>strSrc</code> (optional): the image URL (<code>img.src</code> if undefined)	
Note: you can as well call <code>scanForNewImages()</code> after adding an <code></code> node	

nInstanceID	getInstanceID();
Returns the number of the viewer object.	

strVersion	getVersion();
Returns a string containing the FSI ImageFlow software version.	

strVersion	getBuild();
Returns a string containing the FSI ImageFlow software build number.	

void	removeImages(arImageDOMElements)
Removes one or more quickzoom images and restores their original state.	

void	scanForNewImages();
Calling the scanForNewImages method will make the script look for single source images that have not been present in the DOM tree when the document finished loading. If you e.g. add tags at runtime, you need to call this method in order to make the script aware of new image(s).	

II)Callbacks

void	fnPositionAndSizeCallback();
Returns the position and size.	

fnPositionAndSizeCallbackScope

The scope used when calling fnPositionAndSizeCallback.

Example:

```
function setQuickZoomPos (oModify,oInfo){

    oModify.left    = 0;
    // relative to the left of the image's clientRect
    oModify.top     = 0;
    // relative to the top of the image's clientRect

    oModify.width  = 200;
    oModify.height = 200;
    oModify.srcZoomImage += "&effects=Sepia()";

    // oInfo contains various information you can read
    // modifying oInfo does not change anything
    // e.g.
    // oInfo.elementPosition
    // (clientPosition of the original
    // image tag)
    // oInfo.imgPaddings (margin and border of the image)

}
```

5 FSI ImageFlow

FSI ImageFlow is a Javascript based image carousel with optional image zoom.

Supported devices

- iOS 4 or higher
- Android 4.x based devices
- Windows 8 with touch enabled browsers
- Mac OS with touch device

Usage

I) add the fsiimageflow.js script to the head of your document:

Example:

```
<head>
[...]
<script src="viewer/applications/imageflow/js/fsiimageflow.js"
type="text/javascript">
</script>
```

II) add a <fsi-imageflow> tag to the part of your document body where FSI ImageFlow should be displayed. Add the desired dimension to it and image source (directory) to it:

Example:

```
<body>
[...]
<fsi-imageflow width="100%" height="150" src="/images/foo/" >
</fsi-imageflow>
```

Parameters are added to the <fsi-imageflow> tag. Note that you can as well use configuration files:

Example:

```
<body>
[...]
```

```
//Adding parameters directly to the tag:
<fsi-imageflow width="100%" height="150" src="/images/foo/"
backgroundcolor="#000" presentationtype="flat">
</fsi-imageflow>
```

```
//Example with using a FSI Viewer configuration file:
<fsi-imageflow width="100%" height="150" src="/images/foo/"
cfg="sample/sample_configuration">
</fsi-imageflow>
```

```
</head>
```

5.1 How it works

The script retrieves an image list from FSI Server and display the images in form of an image carousel by means of a `<canvas>` element in the given `<fsi-imageflow>` tag.

Note on Internet Explorer:

In order to make FSI ImageFlow work, Internet Explorer must run in IE8 quirks, IE9 or higher quirks or standards mode. You can ensure that by adding an X-UA-Compatible meta tag to the head section of your web page, for example:

```
<meta http-equiv="X-UA-Compatible" content="IE=Edge,chrome=1">
```

5.2 FSI ImageFlow Parameters

You can optionally modify the way FSI ImageFlow works by passing parameters to the script. You can either use an XML configuration passing the path using a "cfg" parameter or pass parameters and values as an object to the constructor:

Example:

```
<script type="text/javascript">
flow = new $FSI.ImageFlow({debug:true,
backgroundColor:"#ff0"});
flow.init();
</script>
```

debug	
Description	displays debug & status information in the javascript console.
Syntax	Boolean
Default	false
Context	FSI ImageFlow

Display debug and status information in the browser's JavaScript console.

backgroundColor	
Description	The background color for the image carousel.
Syntax	String
Default	"#FFFFFF"
Context	FSI ImageFlow

The background color for the image carousel.

Use "none" if the background should be visible.

Use a web color (e.g. #FFFF00) to define the color of the background.

headers	
Description	display debug and status information in the JavaScript console
Syntax	String
Default	"cors"
Context	FSI ImageFlow

The "&headers=" parameter to use for image requests.
The default value "cors" ensures cors compatibility for cross-origin scenarios.

dir	
Description	The path to the source image folder on FSI Server to display.
Syntax	string
Default	""
Context	FSI ImageFlow

The path to the source image folder on FSI Server to display.

imagesources	
Description	paths to single images used for FSI ImageFlow
Syntax	String
Default	""
Context	FSI ImageFlow

Alternative way to define the paths to single images used for FSI ImageFlow.
Example: "dir1/1.jpeg, dir1/2.jpeg, dir2/1.jpeg,..."

autoResize	
Description	defines if the images are resized if the instance size changes
Syntax	Boolean
Default	true
Context	FSI ImageFlow

Defines if the images are resized if the instance size changes.

transparency	
Description	activates image transparency
Syntax	Boolean
Default	false
Context	FSI ImageFlow

Activates image transparency if PNGs are used.

centerZoomedImage	
Description	moves the zoomed image to the center
Syntax	Boolean
Default	true
Context	FSI ImageFlow

Moves the currently zoomed image to the center of the carousel.

zoomCenteredImageOnly	
Description	zoom the centered image only
Syntax	Boolean
Default	false
Context	FSI ImageFlow

If activated, only the centered image will be zoomed, otherwise any clicked image will be zoomable.

positionQuantization	
Description	aligns the position to the element width
Syntax	Boolean
Default	false
Context	FSI ImageFlow

Aligns the position to the element width. If deactivated, any position is allowed.

elementWidth	
Description	The maximum width of images in the image carousel.
Syntax	String
Default	10 (%)
Context	FSI ImageFlow

The maximum width of images in the image carousel. The value can be provided in pixels or in percent of the instance width.

Note: The maximum height depends on the container element's height, the mirrorHeight and padding parameters.

elementSpacing	
Description	Spacing between the images in pixel.
Syntax	String
Default	0.25 (%)
Context	FSI ImageFlow

Spacing between the images in pixel. The spacing needs to be either stated in pixels or in percent of the instance width.

Note: If the resulting image height limits the image dimension, there is a spacing even if you define elementSpacing with a value of "0". In this case you either need to:

- a) Increase the container <div> element height
- b) Decrease the value of the elementWidth parameter

endlessScrolling	
Description	defines if endless scroll is activated
Syntax	Boolean
Default	false
Context	FSI ImageFlow

Defines if the endless scroll is activated. If deactivated, the images will not be repeated.

duplicateImagesIfNeeded	
Description	duplicates images if endless scroll is activated
Syntax	Boolean
Default	false
Context	FSI ImageFlow

Duplicates the images if endless scroll is activated and the number of images is not sufficient to fill the scroll bar width.

preloadCount	
Description	The number of images to pre-load.
Syntax	Integer
Default	20
Context	FSI ImageFlow

The number of images to pre-load although they are currently not visible. Possible values are integers greater or equal to 0.

scrollAlphaNormal	
Description	opacity of scroll bar in normal mode
Syntax	Float
Default	0.35
Context	FSI ImageFlow

Defines the opacity of the scroll bar in normal mode.

scrollAlphaActive	
Description	opacity of scroll bar in active mode
Syntax	Float
Default	1.0
Context	FSI ImageFlow

Defines the opacity of the scroll bar in active mode.

scrollPadding	
Description	padding of the scroll bar
Syntax	Integer
Default	0
Context	FSI ImageFlow

Defines the padding of the scroll bar.

scrollHeight	
Description	height of the scroll bar
Syntax	Integer
Default	16
Context	FSI ImageFlow

Defines the height of the scroll bar.

autoRotateSpeed	
Description	Image carousel auto-rotation.
Syntax	Integer
Default	0
Context	FSI ImageFlow

Use numbers greater or smaller than 0 to make the carousel rotate on start with the given speed and direction. Auto-rotation stops as soon as the user interacts with the carousel.

paddingBottom	
Description	Bottom padding in pixel.
Syntax	Integer
Default	0
Context	FSI ImageFlow

Use numbers greater or smaller than 0 to make the carousel rotate on start with the Bottom padding in pixel.

cropValues	
Description	Defines the amount that should be cut off the image.
Syntax	String
Default	""
Context	FSI ImageFlow

Defines the amount that should be cut off the image from each side. Values can be provided as pixel (e.g. "20px") or percent (e.g. "10%") or floating point numbers (e.g. "0.25"). Use 4 values for left, top, right, bottom. Example: 10%,0,10%,0

cropRect	
Description	Defines a crop rectangle
Syntax	String
Default	"0,0,1,1"
Context	FSI ImageFlow

Defines the amount that should be cut off the image from each side. Values can be provided as pixel (e.g. "20px") or percent (e.g. "10%") or floating point numbers (e.g. "0.25").

Use 4 values for left, top, right, bottom.

Note: You cannot use "CropRect" AND "CropValues" parameters at a time. If you use both, "CropRect" will be ignored.

effects	
Description	Optional FSI Server effects to use for all images.
Syntax	String
Default	""
Context	FSI ImageFlow

Optional FSI Server effects to use for all images.

Example: effects:"Sepia()"

imageServerTemplate	
Description	The FSI Server template to request images.
Syntax	String
Default	"jpg"
Context	FSI ImageFlow

The FSI Server template to request images.

Usually this is "jpg" or "png".

The default quality is part of the server template definition.

quality	
Description	JPEG quality level when retrieving JPEG images from the server.
Syntax	Integer
Default	-
Context	FSI ImageFlow

The JPEG quality level when retrieving JPEG images from the server.

paddingTop	
Description	Top padding in pixel.
Syntax	Integer
Default	10
Context	FSI ImageFlow

Top padding in pixel.

mirrorHeight	
Description	The height of the optional mirror effect relative to the image height.
Syntax	Float
Default	0.5
Context	FSI ImageFlow

The height of the optional mirror effect relative to the image height.
 0 turns the mirror effect off.
 1.0 uses the entire image height.

mirrorStartOpacity	
Description	The start opacity for the mirror effect.
Syntax	Float
Default	0.35
Context	FSI ImageFlow

The start opacity for the mirror effect.

1.0 means a very heavy mirror effect starting with 100% reflection of the original image.

backgroundDarkness	
Description	This parameter determines, if images in the back will be darkened.
Syntax	Float
Default	0.3
Context	FSI ImageFlow

This parameter determines, if images in the back will be darkened.

0 turns this effect off.

depthFadeout	
Description	An optional value to determine the depth to clip images.
Syntax	Float
Default	0.2
Context	FSI ImageFlow

An optional value to determine the depth to clip images.

If images appear and disappear abruptly at the sides you might want to decrease the value to e.g. 0.1.

enableScrollBar	
Description	enable or disable the scroll bar.
Syntax	Boolean
Default	true
Context	FSI ImageFlow

This value enables or disables the scroll bar.

scrollColor	
Description	The color to draw the scroll bar.
Syntax	6-digit hex color
Default	#555555
Context	FSI ImageFlow

The color to draw the scroll bar.

curveHeight	
Description	This value determines the y-position of images in the back.
Syntax	Float
Default	0.35
Context	FSI ImageFlow

This value determines the y-position of images in the back. 0 means that all images are equally bottom aligned. Values greater than 0 increase the radius like viewing the images from an elevated position.

imageBorderColor	
Description	defines the border around the images.
Syntax	String
Default	"none"
Context	FSI ImageFlow

Use "none" for no borders around the images.
Otherwise define a 6-digit color, e.g. "#ff0000".

placeholderColor	
Description	The color to draw image space holders.
Syntax	String
Default	"#EEEEEE"
Context	FSI ImageFlow

The color to draw image space holders.
Space holders appear while the actual images data is being loaded.
Use "none" to disable drawing placeholders.

presentationType	
Description	The overall image presentation appearance.
Syntax	Enum
Default	"carousel"
Context	FSI ImageFlow

Selects the overall image presentation (carousel, stacks, flat) appearance in FSI Imageflow.

enableZoom	
Description	Enable or disable zooming the focused image.
Syntax	Boolean
Default	true
Context	FSI ImageFlow

Enable or disable zooming the focused image. The focused image is the last image clicked. If the image is not the image in the center, you therefore need to click twice to zoom.

When zooming an image, FSI ImageFlow adds a <div> element to the document body, holding the zoomed image.

zoomBorderWidth	
Description	Width of the border around zoomed images.
Syntax	Integer
Default	4
Context	FSI ImageFlow

Width of the border around zoomed images.

zoomBackgroundColor	
Description	The CSS background-color property to use for zoomed images.
Syntax	String
Default	"#FFFFFF"
Context	FSI ImageFlow

The CSS background-color property to use for zoomed images. Use "none" for a transparent background.

zoomBorderStyle	
Description	The CSS border-style property to use for zoomed images.
Syntax	String
Default	"solid"
Context	FSI ImageFlow

The CSS border-style property to use for zoomed images.

zoomBorderColor	
Description	The CSS border-color property to use for zoomed images.
Syntax	6-digit hex color
Default	#777777
Context	FSI ImageFlow

The CSS border-color property to use for zoomed images.

zoomFadeColor	
Description	The color to use to fade out behind the zoomed image.
Syntax	6-digit hex color
Default	#FFFFFF
Context	FSI ImageFlow

The color to use to fade out the document behind the zoomed image.

zoomFadeOpacity	
Description	The opacity to use to fade out behind the zoomed image.
Syntax	Integer
Default	80
Context	FSI ImageFlow

The opacity (0 to 100) to use to fade out the document behind the zoomed image.

zIndexZoom	
Description	holding the zoom image to the document body.
Syntax	Integer
Default	1000
Context	FSI ImageFlow

The CSS z-index to add the <div> holding the zoom image to the document body.

autoDestroy	
Description	automatically destroy instances
Syntax	boolean
Default	true
Context	FSI ImageFlow

Automatically destroy instances created with the fsi-imageflow tag upon removing the tag from the document DOM.

5.3 Embedding the viewer at runtime via JavaScript

It is also possible to embed the viewer at runtime via JavaScript.

I) Passing parameters via node attribute:

Example:

```
var node = document.createElement("fsi-imageflow");

node.setAttribute("width", "400");
node.setAttribute("height", "600");
node.setAttribute("src", "images/foo/");
node.setAttribute("debug", "true");

document.body.appendChild(node);

$FSI.initCustomNode(node);
// Important: call this AFTER adding the node to the DOM
```

I) Passing parameters via JavaScript object:

NOTE: In this case any parameters passed via node attribute will be ignored.

Example:

```
var node = document.createElement("fsi-imageflow");
node.style.width = "600px";
node.style.height = "400px";

var parameters =
{"dir" : "images/foo/", "presentationtype":"flat",
 "debug":true};

$FSI.setParameters(node, parameters);
document.body.appendChild(node);

// Important: call this AFTER adding the node to the DOM
$FSI.initCustomNode(node);
```

Note:

`$FSI.initCustomNode (node)` initializes the given node.

When adding multiple viewers you can initialize all new FSI ImageFlow nodes in one go, using this command:

```
$FSI.initCustomTag("fsi-imageflow");
```

or, when using different custom tags (fsi-viewer, fsi-imageflow,...):

```
$FSI.initCustomTags (); to initialize all new FSI custom tags.
```

Destroying Instances

Before finally removing `<fsi-imageflow>` nodes from the DOM, you need to call the `destroy()` method of the node.

This will release all event handlers and free allocated resources.

Example:

```
var node = document.getElementById("myViewer");

node.destroy();

node.parentNode.removeChild(node);
```

5.4 JavaScript Interface

```
FSIImageFlow = new $FSI.ImageFlow();
FSIImageFlow.init(dir:"images/foo");
```

I) Public Methods

void	init(elementID, oParameters, bDebug);
Initializes the FSI Image Flow instance with the given parameters.	

void	printAPI();
Shows all API methods in the console in alphabetical order.	

void	destroy();
Destroys the FSI ImageFlow instance leaving the empty container <code><div></code> only. You can re-use the same container calling <code>FSIImageFlow.init()</code>	

bInitDone	getInitDone();
Returns if the initialization process is done.	

nInstanceID	getInstanceID();
Returns the number of the viewer object.	

strVersion	getVersion();
Returns a string containing the FSI ImageFlow software version.	

strVersion	getBuild();
Returns a string containing the FSI ImageFlow software build number.	

void	assignFSIViewer(iFSIViewerInstance);
Assigns a certain FSI Viewer JS instance to FSI ImageFlow for building FSI Showcase JS.	

void	focusImage(nImage);
Rotate the carousel until the image with the given index is in the foreground.	

void	zoomImage(nImage);
Rotate the carousel until the image with the given index is in the foreground and zoom the given image afterwards.	

void	closeZoom(evt, bImmediate);
Remove the zoomed image if any.	

II] Callback functions

The optional JavaScript callback functions enables developers to react on events fired by FSI ImageFlow. To use a callback you need to implement a corresponding

JavaScript function AND pass the name of the callback function to the FSI ImageFlow instance, e.g.:

Example:

```
function onFSIImageFlowStart (oInstance, idElement, strDir,
nImages){
console.log("The FSI ImageFlow instance with the id " +
idElement + " displays a total of " + nImages + " images".)
FSIImageFlow = new $FSI.ImageFlow();
FSIImageFlow.init(dir:"images/foo",
callBackStart:onFSIImageFlowStart);
```

Note: You can either pass the function or the function name in window scope as a string.

The following callback functions can be used:

	callBackStart(oInstance, idElement, strDir, nImagesTotal);
Called once after loading the image list.	

	callBackClick(oInstance, idElement, nImageIndex, strImagePath);
Called each time the user clicks an image.	

	callBackFocus(oInstance, idElement, nImageIndex, strImagePath);
Called each time the foreground image changes.	

	callBackZoomStart(oInstance, idElement, nImageIndex, urlZoomedImage);
Called each time a zoomed image starts loading.	

	callBackZoomEnd(oInstance, idElement);
Called each time a zoomed image has been closed.	

6 FSI Showcase JS

FSI Showcase JS is a combination of the HTML5 and Javascript based viewers FSI Viewer JS and FSI ImageFlow, for displaying multiple images in a scrollable area with a FSI Viewer JS instance above.

Supported devices

- iOS 4 or higher
- Android 4.x based devices
- Windows Phone 8 and Windows RT devices
- Windows 8 with touch enabled browsers
- Mouse zoom: desktop or laptop computers & all major browsers

Usage

To use FSIShowcase JS, make sure the respective images on your website come from FSI Server and add the following scripts to the <head> section of your website:

Example:

```
<head>
[...]
<script src="applications/viewer/js/fsiviewer.js"
type="text/javascript"></script>
<script src="applications/imageflow/fsiimageflow.js"
type="text/javascript"></script>
</head>
```

FSI Showcase is build of combining the FSI Viewer JS and FSI ImageFlow tags, assigning theViewer JS to ImageFlow with a specific viewer id:

Example:

```
<body>
[...]
```

//Adding FSI Viewer

```
<fsi-viewer id="viewer1" width="100%" height="600" noNav="1"
plugins="fullScreen">
</fsi-viewer>
```

//Adding FSI ImageFlow, assigning the Viewer JS with the viewerID

```
<fsi-imageflow viewerID="viewer1" width="100%" height="150"
src="/images/foo/" presentationType="flat">
</fsi-imageflow>
```

7 FSI Pages JS

FSI Pages JS is a JavaScript based tool for presenting image collections as an interactive catalog, booklet or photo album on desktop computers and mobile devices. FSI Pages JS automatically creates a page index with thumbnails and provide hyperlinks and interactive zooming



7.1 How it works

The script uses the given `<fsi-pages>` tag, which is similar to an `` tag, to display an interactive catalog.

In the tag, a directory is defined which will contain all images displayed. It is possible to add tool tips and hyperlinks to the pages for displaying product descriptions or to link to an online shop. A chapters plug in enables to user to guide through large catalogs easily.

The user can drag the pages left and right to browse through the catalog and zoom in/out either using pinch zoom or by double-clicking on the page.

Hovering over embedded areas displays a tool tip and/ or hyperlink.

7.2 Usage

I) add the `fsipages.js` script to the head of your document:

Adding the script

```
<head>
[...]
//adding the fsipages.js script
<script src="viewer/applications/pages/js/fsipages.js"
type="text/javascript"></script>
</head>
```

II) Add a `<fsi-pages>` tag to the part of your document body where the catalog should be displayed. Add the desired dimension and the image directory to it:

Integrate FSI Pages JS

```
<body>
[...]
<fsi-pages width="100%" height="100%" dir="/images/catalog/">

//here you can place the fallback content (or image) in case the
browser
// cannot display the viewer, e,g,:
<b>Sorry</b>, your browser is not capable of running FSI Pages JS

</fsi-pages>
```

Parameters are added to the `<fsi-pages>` tag. Note that you can as well use FSI Pagesr configuration files :

Adding Parameters to FSI Pages JS

```
<head>
[...]
//adding the fsipages.js script
<script src="viewer/applications/pages/js/fsipages.js"
type="text/javascript"></script>
</head>

<body>
[...]

//Adding parameters directly to the tag:
<fsi-pages width="500" height="600" dir="/images/catalog/"
debug="true" skin="silver" PageLayout="flip">
</fsi-pages>

//Example with using a FSI Pages configuration file:
<fsi-pages width="500" height="600" dir="/images/catalog/"
cfg="sample/catalog.xml">
</fsi-pages>
```

7.2.1 Defining Image Collections

The images to be displayed in FSI Pages JS have to be defined in an `<images>` node added to your XML configuration file.

There are three different ways of defining image lists:

1. Retrieving automatically generated image lists from FSI Server : The most convenient way to create image collections is using the template system of FSI Server. You can either do so by using the "Publish to Web" option in FSI Server web interface or by specifying the server address and a directory or search query manually.

Retrieving Image Collections from Imaging Server

```
<PLUGIN src="pages">
  <ListTemplate value="catalog_list" />
  <dir value="images/catalogs/foo" />
</PLUGIN>
```

This way FSI Server uses an image → *ListTemplate* to return the matching images in XML format. Depending on the list template, FSI Server serves IPTC data belonging to an image (e.g. for links) as well.

2. Retrieving Directory based Image Collections from FSI Server

Retrieving Directory based Image Collections from FSI Server

```
<Plugins>
...
  <!-- Passing a directory as plain text -->
  <Plugin src="pages" server="..."
    dir="images/flowers/" />
  <!-- Passing an encrypted string representing a
    directory -->
  <Plugin src="pages" server="..."
    dir="*XzPzLUFVSHtmeeFxOY0NUU0" />
...
</Plugins>
```

3. Creating collections manually using XML configuration files

Creating collections manually using XML configuration files

```
<Images>
...
  <Image file="image1.xml"
    label="My First Image" />
  <Image file="image2.xml"
    label="My Second Image" />
...
</Images>
```

Method 2 and Method 3 can be freely combined while Method 1 retrieves a complete image list ignoring previously defined <images> nodes.

Image Order

The order of the images (this is the page order) corresponds to the order of the `<image>` declarations within the `<images>` section.

Inserting Blank Pages

You can insert blank pages by adding `<image empty="true" />` nodes anywhere in your image collection. Alternatively you can use the FSI Pages parameter `→ emptyImages` to pass the page numbers to insert blank pages before separated by commas. This is especially useful if you retrieve the image collection from FSI Server (Method 1).

Removing certain Images from a List

Using the parameter `→ removePages` you can disable (remove) certain images from an image list. In this case you pass the page number(s) of the pages you want to remove separated by commas.

7.2.2 Aspect Ratio of the Pages

By default FSI Pages uses the aspect ratio of the first image in the image collection to determine the aspect ratio for all pages.

Alternatively you can manually define an aspect ratio for all pages using the `"ratio"` parameter.

E.g. `ratio="1:2"`
or `ratio="640:480"`

7.2.3 Printing Pages

If you want to enable users to print pages from within FSI Pages you can enable the built-in print function by setting the FSI Pages JS parameter `→ Print` to `"true"`.

Enabling Print Options

```
<Plugins>
...
  <Plugin src="pages" >
    <Print value="true" />
    <PrintResolution value="1024" />
  </Plugin>
...
</Plugins>
```

This adds a "Print" button to the menu bar. Clicking the "Print" button displays the "Print" dialog of FSI Pages JS. The → *Print* dialog offers different printing options to the user, depending on the current page and if the user zoomed into a page.

The printing options cover:

- Selected Range (e.g. 1-3; 9-20 or 1,6,8,12)
- All pages currently in view
- All pages

After the user has chosen an option and clicks the "Print" button in the dialog, FSI Pages JS downloads the required image data (pixel data) from the imaging server and opens the print dialog on the user's system once the download has finished. You can limit the resolution used for printing using the FSI Pages JS parameter → *PrintResolution*. Lower resolution leads to faster downloads at the cost of decreased print quality.

You can additionally pass effect parameters → *PrintEffects* to sharpen the image or to specify the image compression.

7.2.4 Special URL Values

For some special URL values FSI Pages JS ignores the URLPrefixes and URLSuffixes as well. The following table lists these special URLs:

URL	Description
javascript:foo()	Call the JavaScript function foo()
mailto:email@domain.tld	Send an email to email@domain.tld
#none	Do nothing. Handy if you just want to display a tool tip. Available in FSI Pages version 4.0.7 or above
#page=n	Forces FSI Pages to flip to page n
#showimage=[imageURL]	Display a custom JPEG image. The value [URL] needs to be an absolute URL to a JPEG image.
#zoom	Forces FSI Pages to zoom to the area covered by the link's shape(s)
#zoomarea	Magnifies the area covered by the link's shape(s) in place
#zoom=1,1,0,0,0.5,0.5	Forces FSI Pages to zoom to the image area

URL	Description
	following "=" (See Parameter → <i>InitialView</i> for format details)
#zoomonpage=n;strView	Combination of "#page=" and "#zoom=". This is: first flip to page n, then zoom to area strView. Available in FSI Pages version 4.0.7 or above

Please note that the special links starting with "#" need to be provided exactly as listed in the table above. Using e.g. "#zoomOnPage=[...]" or "foo.html#zoom" will link to the HTML anchor provided rather than executing an FSI Pages JSaction.

For "mailto:" URLs the target frame is always "_self", except the XML <area> node specifies a different target.

7.2.5 FSI Pages JS Presets in FSI Server Interface "Publish to Web"

You can choose from the following presets when using the "Publish as FSI Pages" option in the FSI Server web interface:

- **Catalog with links** Each image represents a page, no page margins, no overlays. If the image contains XML data in the "IPTC_FSI" data field, links will be presented on the page.
- **Catalog without links** Each image represents a page, no page margins, no overlays.
- **Teaser Thumbnail** No user interface, zoom is disabled. Pages flip automatically when idle. You can specify a link for the entire viewer area so that you can use this template to publish a thumbnail of a catalog linking to the full size catalog.

7.3 FSI Pages JS Parameters

You can optionally modify the way FSI Pages JS works by passing parameters to the script. To do so, pass an object containing the parameters and value as an object to the constructor:

Example:

```
<head>
[...]
//adding the fsipages.js script
<script src="viewer/applications/pages/js/fsipages.js"
type="text/javascript"></script>
</head>

<body>
[...]

//Adding parameters directly to the tag:
<fsi- pages width="500" height="600" dir="/images/catalog/"
debug="true" skin="silver" PageLayout="flip" Links="false">
</fsi-pages>
```

7.3.1 General Parameters

debug	
Description	display debug and status information in the JavaScript console
Syntax	Boolean
Default	false
Context	FSI Pages JS

Display debug and status information in the browser's JavaScript console.

skin	
Description	defines the skin to display
Syntax	string
Default	"black"
Context	FSI Pages JS

Defines the skin which will be displayed. FSI Pages JS comes with three default skins, "black", "white" and "silver".

Skins are defined via CSS and can be found here: [WEBAPPS/fsi/viewer/skins](#).

It is also possible to alter existing skins or to define custom skins in this location.

In order to use a custom skin, the easiest way would be to duplicate one of the default skin CSS files and adapt them to your liking. make sure to use a custom class in front of ALL classes defined in the custom CSS,, otherwise you will break other skins on the same HTML document.

General appearance CSS rules, which are valid on an overall level, (e.g. the logo container, height of the menubar, etc.) are defined in the corresponding core CSS files, which can be found in WEBAPPS/fsi/viewer/skins/resources. (fsi-core.css, fsi-viewer-core.css)

IMPORTANT: Never change the classes of the core CSS in the core files directly. If you would like to adapt the CSS rules listed there, please overwrite them in a custom skin file or in the HTML document instead.

You can find detailed tutorials for changing the appearance of the skin in the *support section of our website*.

For a detailed example of a basic custom skin, please go to→ "*Example of a custom skin CSS file*".

dir	
Description	path to the source images for the catalog on FSI Server
Syntax	String
Default	""
Context	FSI Pages JS

The path to the source images for the catalog on FSI Server.
The viewer will load the list of images from the given directory.

language	
Description	User interface language
Syntax	String
Default	""
Context	FSI Pages JS

Defines the language of tool tips for the user interface (menu bar).
Possible values depend on the XML based language files located in the sub-index

"/languages" of the FSI Viewer setup path. To specify a language, please enter the filename without file extension, e.g. "german".

You can modify existing or add your own "language files".

cfg	
Description	relative path to an XML configuration file in FSI Pages format
Syntax	String
Default	""
Context	FSI Pages JS

The relative path to an XML configuration file in FSI Pages format.

hideUI	
Description	hides the user interface
Syntax	string
Default	false
Context	FSI Pages JS

Hides or displays the user interface.

Possible values:

true

false

onMobileDevices: UI is hidden on mobile devices

ifNoInPlaceZoom: UI is hidden if in place zoom is deactivated

adaptiveUISize	
Description	determines if interface scales according to the zoom level .
Syntax	Boolean
Default	true
Context	FSI Pages JS

When set to true, the interface scales according to the zoom level.

Zoom	
Description	Enables/ disables page zoom
Syntax	Boolean
Default	true
Context	FSI Pages JS

Enables/ disables page zoom.

MaxZoom	
Description	Maximum zoom in percent of the source image.
Syntax	Float
Default	100
Context	FSI Pages JS

Maximum zoom in percent of the source image.

MenuButtonOrder	
Description	determines the button order in the user interface.
Syntax	string
Default	
Context	FSI Pages JS

Determines the button order in the menu bar. Use the button ID which can be found in the respective skin.css (viewer\skins\resources\fsi-pages-core.css) for listing them. Example MenuButtonOrder="ToggleFullscreen, FirstPage, LastPage"

Currently the following button IDs exist:

PrintPage, FirstPage, PreviousPage, PageInputEnter, NextPage, LastPage, ToggleBookmarkList, ToggleBookmark, ToggleTableOfContents, TogglePageIndex, ToggleFullScreen

ButtonsFirstLastPage	
Description	Enables/ disables First Page and Last Page Buttons
Syntax	Boolean
Default	true
Context	FSI Pages JS

When set to true, the buttons for directing to the first or the last page appear.

Index	
Description	Enables / disables page overview
Syntax	Boolean
Default	true
Context	FSI Pages JS

Enables or disables the page index containing thumbnails of all pages.

PageInput	
Description	Enables / disables page input field
Syntax	Boolean
Default	true
Context	FSI Pages JS

Specifies whether a text box for the page input is displayed in the menu bar.

PageInputEnterButton	
Description	Enables / disables page overview
Syntax	Boolean
Default	true
Context	FSI Pages JS

Specifies whether the enter button for the page input field is displayed in the menu bar.

ListTemplate	
Description	XML based template to use when retrieving image lists from the image server
Syntax	String
Default	"image_list.xml"
Context	FSI Pages JS

The XML based template to use when retrieving image lists from the image server via "dir" or "query" parameters. Available templates are located in the "WEB-INF/templates/fsi/" directory of your server setup.

PageDataTemplate	
Description	XML based template to use when
Syntax	String
Default	
Context	FSI Pages JS

The XML based template to use when

DemoURL	
Description	Global link when clicking FSI Pages JS
Syntax	String
Default	-
Context	FSI Pages JS

Specifies a link to open when the user clicks anywhere on the FSI Pages JS instance. This parameter is especially useful when using a small preview of a catalog to lead the user to the full size version of FSI Pages JS. You can use URLs starting with "javascript:" to have a Javascript function called when the user clicks the viewer.

DemoURLTarget	
Description	HTML frame in which DemoURL is opened
Syntax	String
Default	-
Context	FSI Pages JS

HTML frame in which to open the URL specified by "DemoURL" parameter in. You can use default frame names like "_self" or "_blank" or use a custom frame of your HTML frame set.

Print	
Description	Enables or disables the user to print
Syntax	Boolean
Default	true
Context	FSI Pages JS

Enable or disable the user to print current pages, the complete catalog or specific pages from the image collection shown with FSI Pages JS.

PrintResolution	
Description	Specifies the print resolution
Syntax	Integer
Default	2000
Context	FSI Pages JS

Specifies in which resolution the user is allowed to print the pages.

PrintEffects	
Description	Specifies print effects
Syntax	String
Default	
Context	FSI Pages JS

Defines image modification parameters to apply to images used for printing. You can sharpen the image and define the JPEG compression level, e.g. "sharpen=100&quality=95".

7.3.2 Layout/ Appearance Parameters

PageLayout	
Description	Animation when moving to another page
Syntax	String
Default	"scroll"
Context	FSI Pages JS

Defines the animation when moving to another page. "scroll" slides the pages as if they were positioned next to each other (like in FSI Pages mobile). "flip" shows a page turn effect.

InitialPage	
Description	Page to display on startup
Syntax	Integer
Default	1
Context	FSI Pages JS

Specifies the page to display on startup, e.g. "1" for the front cover.

Ratio	
Description	Page to display on startup
Syntax	String
Default	1:1
Context	FSI Pages JS

The aspect ratio of pages displayed in FSI Pages.
You can specify any aspect ration (e.g. "320:240" or "1:2") or specify "auto" to use the aspect ratio of the first image in the image collection.

cropRect	
Description	Defines a crop rectangle
Syntax	String
Default	"0,0,1,1"
Context	FSI Pages JS

Crop rectangle (left, top, right, bottom) as 0.0..1.0 floating point values to display a given section of an image only.
Default: "0,0,1,1"

Note:

You cannot use "CropRect" AND "CropValues" parameters at a time.
If you use both, "CropRect" will be ignored.

cropValues	
Description	Defines crop margin values for a page
Syntax	String
Default	"0,0,0,0"
Context	FSI Pages JS

Defines the amount that should be cut off the image from each side.
Values can be provided as pixel (e.g. "20px") or percent (e.g. "10%") or floating point numbers (e.g. "0.25").

effects	
Description	Image manipulation parameters for all pages
Syntax	String
Default	
Context	FSI Pages JS

The effects parameter can be used to pass image manipulation parameters to the imaging server. Typical manipulation parameters include the image saturation ("saturation") and image sharpening ("sharpen"). If no effect is specified, the default settings of the server (profile) apply. Please refer to your FSI Server documentation for a list of available effect parameters.

ThumbSize	
Description	Max. size of thumbnails in the page index
Syntax	Integer
Default	160
Context	FSI Pages JS

The maximum size of thumbnails displayed in the index. The actual width and height depend on the aspect ratio of the pages.

Parameters only valid if PageLayout is set to scroll:

verticalLayout	
Description	moves pages vertical
Syntax	Boolean
Default	"false"
Context	FSI Pages JS

When set to true, the pages scroll in the vertical direction instead of the usual horizontal.

Note: only valid if PageLayout is set to "scroll"!

onePageInView	
Description	sets only one page in view
Syntax	Boolean
Default	"false"
Context	FSI Pages JS

Note: only valid if PageLayout is set to "scroll"!

doublePageMode	
Description	enables double page view
Syntax	Boolean
Default	"true"
Context	FSI Pages JS

If set to "false", the pages will not be displayed as coherent double pages.

Note: only valid if PageLayout is set to "scroll"!

autoDoublePageMode	
Description	
Syntax	Boolean
Default	"true"
Context	FSI Pages JS

Note: only valid if PageLayout is set to "scroll"!

bendEffect	
Description	gives pages a glossy bend effect
Syntax	String
Default	"glossy"
Context	FSI Pages JS

Defines if the pages are shown with a bend effect. "none" disables the effect.

FrontCover	
Description	enables/ disables the front cover
Syntax	Boolean
Default	"true"
Context	FSI Pages JS

Setting "FrontCover" parameter to false disables access to the front cover and forces FSI Pages to display the first image on the first inner left page.

BackCover	
Description	Enables/disables the back cover
Syntax	Boolean
Default	"true"
Context	FSI Pages JS

Setting "BackCover" parameter to false disables access to the back cover and forces FSI Pages to add a blank content page if required.

FrontCoverImage	
Description	Path to an image to use for the front cover
Syntax	String
Default	-
Context	FSI Pages JS

A path to an image to use for the front cover. FSI Pages JS will use the "ImageServer" parameter of FSI Viewer for relative paths. The cover page(s) will be added to the images in the image list.

BackCoverImage	
Description	Path to an image to use for the back cover
Syntax	Boolean
Default	"true"
Context	FSI Pages JS

A path to an image to use for the back cover. FSI Pages JS will use the "ImageServer" parameter of FSI Viewer for relative paths. The cover page(s) will be added to the images in the image list.

BlankBackCover	
Description	Use a blank page for back cover
Syntax	Boolean
Default	"true"
Context	FSI Pages JS

Use a blank page as back cover.

emptyImages	
Description	add blank pages
Syntax	String
Default	
Context	FSI Pages JS

Using this parameter you can add blank pages. The value of the parameter must contain one or more image page numbers of the blank pages to be inserted. E.g. `<EmptyImages value="2,10" />` adds two blank pages at page 2 and page 10, moving the other images in your collection correspondingly. Alternatively you can add `<image empty="true" />` nodes to your image collection.

removePages	
Description	Removes pages from the image collection
Syntax	String
Default	
Context	FSI Pages JS

Using this parameter you can remove images from the image collection. The behavior of FSI Pages is exactly as if the images would not be listed in the image collection. E.g. `<EmptyImages value="1,3,4" />` removes the first, third and fourth image in the collection. Please note that removing images takes place before inserting blank pages using the "EmptyImages" parameter.

IdleAutoTurn	
Description	Turns pages automatically when idle
Syntax	Boolean
Default	false
Context	FSI Pages JS

Run the FSI Pages JS in demonstration mode and turn pages automatically when the user does not interact (= moves the mouse) for 2 seconds.

IdleAutoTurnMinPage	
Description	Defines the range for automatical page turning
Syntax	Integer
Default	-
Context	FSI Pages JS

Loop the auto flip action between IdleAutoTurnMinPage and IdleAutoTurnMaxPage. This way you can restrict the auto flip to a range of pages. If the initial page is outside the range specified, FSI Pages flips towards the range before looping between the range specified.

IdleAutoTurnMaxPage	
Description	Defines the range for automatical page turning
Syntax	Integer
Default	-
Context	FSI Pages JS

Loop the auto flip action between IdleAutoTurnMinPage and IdleAutoTurnMaxPage. This way you can restrict the auto flip to a range of pages. If the initial page is outside the range specified, FSI Pages flips towards the range before looping between the range specified.

IdleAutoTurnDelay	
Description	Specifies how long pages are shown when IdleAutoTurn is activated
Syntax	float
Default	2
Context	FSI Pages JS

Show each page for at least n seconds when flipping pages automatically. Please note that a page might be displayed for longer if loading the page takes longer than the time specified.

PageNumbers	
Description	Display page numbers on pages
Syntax	Boolean
Default	false
Context	FSI Pages JS

Show or hide page numbers on pages.

FirstPageNumber	
Description	Defines with which page number the catalog starts
Syntax	Integer
Default	1
Context	FSI Pages JS

Defines with which page number the catalog starts. By default (FirstPageNumber=1) the front cover is page number one.

You can specify an offset so that the page numbers start with a different value.

This parameter affects page numbers, the index and the display of the current page number in the FSI Pages JS user interface.

UseRomanPageNumbersToPage	
Description	Display Roman page numbers up to page number n
Syntax	Integer
Default	0
Context	FSI Pages JS

Defines the page number up to which FSI Pages JS uses Roman page Numbers. The default value of "0" forces FSI Pages JS to display Roman numbers for negative numbers and 0.

If you set the value to "5" the display will be like this:
I,II,III,IV,V,6,7,8,9...

Note that you can use the Parameter → *FirstPageNumber* to offset all page number values.

RomanPageNumbersOffset	
Description	Offset Roman page numbers by given value
Syntax	Integer
Default	0
Context	FSI Pages JS

Defines the value of the first Roman page number.
The default value of "0" means no offset so that the Roman numbers start with "I".

CustomPageNumbers	
Description	Comma separated string of custom page numbers
Syntax	String
Default	-
Context	FSI Pages JS

Use "CustomPageNumbers" to specify custom page numbers as a comma separated string. Items starting with underscore ("_") will not be displayed.
Pages without a custom page number will use the ordinary page numbering.

Example: `< CustomPageNumbers value="_cover1,_cover2,3,4,5,_" />`

In this case the cover page and the first left page will have no page number.
The first right page will start with page number "3". The sixth page again has no page number and the seventh and following page will be numbered as usual.

Note: The parameter `→ FirstPageNumber` has no impact on the custom page numbers.
You can therefore use this parameter to modify the value of the first ordinary page number.

CustomPageNumbersFile	
Description	URL or path to an XML file containing custom page numbers
Syntax	String
Default	-
Context	FSI Pages JS

Same as CustomPageNumbers, but the page number values will be loaded from an XML file. You can either define a path relative to the fsi/config/pages_xml/ directory or an absolute URL. The required format of the XML file is as follows:

```
<fsi>
<pagenumbers>3,6,7</pagenumbers>
</fsi>
```

RememberLastViewedPage	
Description	stores page number and displays page on startup next time
Syntax	Boolean
Default	false
Context	FSI Pages JS

If you enable this parameter FSI Pages stores the page number of the most recently viewed page on the user's computer and displays this page on start up the next time the user views the same catalogue (image collection) again.

RememberLastViewedPageExpireAfter	
Description	defines how long last viewed page will be stored
Syntax	Interger
Default	-
Context	FSI Pages JS

By default the last viewed page will be restored on the next visit of a user if you enable the RememberLastViewedPage parameter. You might want to specify an expire time after that the last viewed page will be discarded. Please specify the expire time in seconds, e.g. "3600" for one hour.

By default, the stored pages will be stored infinitely.

fsi-pages-intro-content	
Description	Defines content to be displayed next to front cover
Syntax	String
Default	-
Context	FSI Pages JS

You can define some HTML Code to be displayed next to the front cover of the publication. This is useful to display e.g. usage instructions.

The content needs to be defined in the <fsi-pages> tag as following:

Example:

```
<head>
[...]
//adding the fsipages.js script
<script src="viewer/applications/pages/js/fsipages.js"
type="text/javascript"></script>
</head>

<body>
[...]
<fsi-pages width="500" height="600" dir="/images/catalog/"
debug="true" skin="silver" PageLayout="flip" Links="false">

//Adding intro content:
<fsi-pages-intro-content>
<h3>Welcome to our catalog!</h3>
<br />
<ul>
<li>Tap the Next Page/Previous Page buttons </li>
<li>Double-click or pinch-zoom on a page to zoom in</li>
<li>Tap the full-screen icon to switch to full-screen mode</li>
<li>Click on the index icon for a catalog overview</li>
<li>Enter a number in the page input field to browse to a certain
page</li>
</ul>
</fsi-pages-intro-content>
</fsi-pages>
```

7.3.3 Link Parameters

Links	
Description	Enables/disables links
Syntax	Boolean
Default	true
Context	FSI Pages JS

Specifies whether links are displayed. If set to "false" Links and tool tips will not be displayed.

linkRGBANormal	
Description	Specifies color & opacity of links in normal state
Syntax	String
Default	"0000FFFF"
Context	FSI Pages JS

8-digit hexadecimal number specifying the color and opacity of links on the pages in normal state in the form "RRGGBBAA".

linkRGBAMHover	
Description	Specifies color & opacity of links in hover state
Syntax	String
Default	"FF00FFFF"
Context	FSI Pages JS

8-digit hexadecimal number specifying the color and opacity of links on the pages in hover state in the form "RRGGBBAA".

linkRGBAAActive	
Description	Specifies color & opacity of links in active state
Syntax	String
Default	"FF00FFFF"
Context	FSI Pages JS

8-digit hexadecimal number specifying the color and opacity of links on the pages in active state in the form "RRGGBBAA".

FollowLinks	
Description	Specifies if links are opened
Syntax	Boolean
Default	true
Context	FSI Pages JS

Setting this parameter to "false" prevents FSI Pages JS from opening links on the pages. Links and tool tips will nevertheless be displayed.

LinkTemplates	
Description	Template identifiers for link URLs
Syntax	String
Default	-
Context	FSI Pages JS

You can provide one or more identifiers that will be replaced in Link URLs. The identifiers are case sensitive and must be separated by commas. The values to insert have to be defined by the parameter "LinkTemplateData" explained below.

LinkTemplateData	
Description	Values to replace template identifiers in link URLs
Syntax	String
Default	-
Context	FSI Pages JS

When defining "LinkTemplates" to replace place holders in Link URLs you can use this parameter to specify the values that shall be inserted. The number and sequence of the values must match the number and sequence of the templates you defined using the "LinkTemplates" parameter.

Multiple values have to be separated by commas and the individual values have to be provided url-encoded.

ForceLinkURL	
Description	Force URL for all links on pages
Syntax	String
Default	-
Context	FSI Pages JS

Force FSI Pages JS to use this URL for all links on all pages. This overwrites any URLs specified in the page data.

ForceLinkUrlPrefix, ForceLinkUrlSuffix	
Description	Force a link prefix / suffix for all links on pages
Syntax	String
Default	-
Context	FSI Pages JS

Force FSI Pages JS to use this prefix /suffix for all links on all pages. This overwrites any prefixes and suffixes specified in the page data.

DefaultLinkURL	
Description	Default URL for links on pages
Syntax	String
Default	-
Context	FSI Pages JS

Use this URL for links if no URL has been specified in the page data.

DefaultLinkUrlPrefix, DefaultLinkUrlSuffix	
Description	Default prefix / suffix for link URLs on pages
Syntax	String
Default	-
Context	FSI Pages JS

Use this prefix or suffix for all links on the pages if no prefix / suffix has been specified in the page data.

DefaultLinkTarget	
Description	Default target frame for all hyperlinks on all pages
Syntax	String
Default	-
Context	FSI Pages JS

Use this HTML target frame for links if no target frame has been specified in the page XML data.

ForceLinkTarget	
Description	Force a target frame for all hyperlinks on all pages
Syntax	String
Default	-
Context	FSI Pages JS

Force FSI Pages JS to use this HTML target frame for all links on all pages. Example. "_ blank"

This overwrites any target frames specified in the page XML data.

Note: this parameter does not affect "javascript:" and "mailto:" links.

ForceJavascriptTarget	
Description	Force a target frame for all "javascript:" links on pages
Syntax	String
Default	-
Context	FSI Pages JS

Force FSI Pages JS to use this HTML target frame for all "javascript:" links on all pages.

This overwrites any target frames specified in the page XML data.

Note: this parameter does not affect URL and "mailto:" links.

DefaultJavascriptTarget	
Description	Default target frame for all "javascript:" hyperlinks on all pages
Syntax	String
Default	-
Context	FSI Pages JS

Use this HTML target frame for "javascript:" links if no target frame has been specified in the page data.

ForceLinkTip	
Description	Force a tool tip for all links on pages
Syntax	String
Default	-
Context	FSI Pages JS

Force FSI Pages to use this tool tip for all links on the pages. This overwrites any tool tips specified in the page data.

DefaultLinkTip	
Description	Default tool tip for links on pages
Syntax	String
Default	-
Context	FSI Pages JS

Use this tool tip for all links on the pages if no tool tip has been specified in the page data.

7.3.4 Plug-in Parameters

plugins	
Description	adds a Plug-in (e.g. Fullscreen) to FSI Pages JS
Syntax	String
Default	-
Context	FSI Pages JS

Adds a plug-in to the viewer. Multiple plug-ins can be listed separated by comma.

Please note:

If you would like to add specific parameters to the plugins via the fsi-pages tag, you can do this by prefixing the parameter name with the name of the plug-in, e.g. `pagesthumbbar_height="100"`

FullScreen

Adding the parameter `plugins:"FullScreen, Resize"` enables a full screen view option in FSI Pages JS. The `Resize` parameter needs to be set in order to ensure proper adjusting to the full screen.

NOTE: If the FSI Pages JS is embedded in an `iFrame`, the attribute `allowfullscreen="true"` needs to be added to the `iFrame`. Otherwise the `FullScreen` Plugin will not work due to security restrictions of the browser.

PagesThumbBar

Adding the parameter `plugins:"PagesThumbBar"` displays a row of thumbnails at the bottom of the pages.

The following parameters can be used to adapt the `PagesThumbBar`:

PageNumbers	
Description	Page numbers being displayed below thumbnails
Syntax	Boolean
Default	true
Context	FSI Pages JS

Enables/ disables the display of page numbers being below the thumbnails.

Autohide	
Description	Auto hides thumbar below 600px
Syntax	Boolean
Default	true
Context	FSI Pages JS

If set to true, the thumbnails will not be displayed if the FSI Pages Instance is smaller than 600px.

Height	
Description	Defines height of the thumbnails
Syntax	Integer
Default	160
Context	FSI Pages JS

Defines the height of the thumbnails displayed.

Bookmarks

Adding the parameter `plugins: "Bookmarks"` adds a bookmark functionality to FSI Pages JS.

The following parameters can be used to adapt the bookmarks functionality:

ThumbSize	
Description	Defines the thumbnail size in the bookmark list.
Syntax	Float
Default	200
Context	FSI Pages JS

Defines the maximum thumbnail size in the bookmark list in pixel.

ShowIconsOnPages	
Description	Shows icon on bookmarked pages
Syntax	Boolean
Default	true
Context	FSI Pages JS

Shows icon on bookmarked pages.

PageNumbers	
Description	Shows page numbers in bookmark list
Syntax	Boolean
Default	true
Context	FSI Pages JS

Shows page numbers in bookmark list

PersistentStorage	
Description	Defines if the bookmark data is being cached..
Syntax	Boolean
Default	true
Context	FSI Pages JS

When disabled, the bookmarks will be deleted when the user leaves the FSI Pages instance.

When enabled the data will be cached for the given time (defined by PersistentStorageExpiresAfter)

PersistentStorageExpiresAfter	
Description	Keep note data for given time.
Syntax	Integer
Default	-
Context	FSI Pages JS

Defines after which period of time the cached note data expires.

Chapters

Adding the parameter `plugins: "Chapters"` adds a select box to the user interface of FSI Pages JS. The user can directly access sections of a catalog by selecting the corresponding item in the select box. You can assign colors and different levels to each chapter using simple XML based data describing the document structures.

Example:

```

<Indexdata>
<Index label="First Page" page="1" color="FFFF00" />
<Index label="Chapter 1" page="10" color="00FF00">
<Index label="Chapter 1.1"
page="20" color="00FF00"/>
<Index label="Chapter 1.2"
page="30" color="00FF00"/>
<Index label="Chapter 1.3"
page="40" color="00FF00"/>
</Index>
<Index label="Chapter 2" page="50" color="0000FF"/>
<Index label="Chapter 3" page="60" color="CCCCCC"/>
<Index label="Chapter 4" page="70" color="FF0000"/>
</Indexdata>

```

The following parameters can be used to adapt the Chapters:

indexDataFile	
Description	URL to an external XML file describing the document structure
Syntax	string
Default	-
Context	FSI Pages JS

You can use an external XML file to provide the XML document structure data required for this plug-in. The XML file must be located in the same domain as FSI Viewer to avoid cross-domain security restrictions.

Width	
Description	Width of the select box
Syntax	Integer
Default	-
Context	FSI Pages JS

Usually the FSI Pages JSskin defines the width and position of the select box. Alternatively you can specify a fixed width in pixels using this parameter.

7.4 JavaScript Interface

The JavaScript interface of FSI Pages JS can be useful if you - for example - want to set certain actions when FSI Pages JS is initialized, while pages change, the user interface is used or links are clicked.

7.4.1 Public Methods

void	start();
Starts FSI Pages JS.	

nID	getInstanceID();
Returns the ID of the Pages JS object.	

void	changeConfig(strCfgFileName, oParameters);
<p>Reset the viewer object and change the configuration to the given configuration file.</p> <p><code>strCfgFileName</code> (optional): path to the configuration file (see "cfg" parameter)</p> <p><code>oParameters</code> (optional): an object containing parameters. If present, these parameters will overwrite the parameters defined upon initialization.</p>	

void	printAPI();
Shows all API methods in the console in alphabetical order.	

void	init(domElement, oParameters, bDebug);
Initializes the chosen Pages JS element.	

bInitDone	getInitDone();
Returns if the initialization process is done.	

void	destroy();
Destroys the given FSI Viewer JS object. You should destroy the instance before you remove the object spins <div> tag from the DOM tree.	

bSuccess	addListener(strListenerName, fn, iScope);
Calls a specific listener..	

bSuccess	removeListener(strListenerName, fn);
Removes a specific listener..	

void	showAboutWindow();
Shows the "About" window.	

void	nextPage();
Goes to next page.	

void	previousPage();
Goes to previous page.	

void	firstPage();
Goes to first page.	

void	lastPage();
Goes to last page.	

bPresent	getMenuButtonPresent(strButtonID)
Shows active menu buttons, can be used to remove inactive buttons from the menu if you are using a custom skin.. NOTE: Method only delivers correct results after the callback "onInitMenu" is called.	

void	pressMenuButton(strButtonID)
Sets the menu button to active.	

void	clickMenuButton(strButtonID)
Sets the menu button to active.	

void	releaseMenuButton(strButtonID)
Disables the active menu button.	

void	registerExternalMenuButton(elButton, strButtonID, strTip)
Registers an external menu button.	

void	unregisterExternalMenuButton(elButton)
Unregisters an external menu button.	

void	registerExternalMenuButtonsFromContainer(elContainer)
Registers an external menu button from a specific container.	

void	unregisterExternalMenuButtonsFromContainer (elContainer)
Unregisters an external menu button from a specific container.	

strClassName	getSkinClassName()
Returns the name of the current used skin class, e.g. fsi-skin-black	

bFullscreen	getInFullscreenMode()
Enables the FullScreen mode.	

bSuccess	setFullScreenElement(elFullScreenContainer)
Defines the FullScreen element.	

strPageNumber	getCurrentPageNumber()
Returns the current page number.	

strPageNumber	getPageNumberString(nPage)
Returns the current page number.	

oDimension	getPageContainerDim()
Returns the Page container dimensions.	

void	goToPage(strPage, bAnimate)
Moves to a certain page. strPage is a string and works like the parameter → <i>PageInput</i> .	

void	goToPageNumeric(nPage, bAnimate)
Moves to a certain page. Works like goToPageNumber, but with nPage being numeric.	

nPage	getTargetPage(nPage)
Return the nPage limited to minPage and maxPage. If FSI Pages shows double pages, the value of the left page matching nPage will be returned.	

strVersion	getVersion();
Returns a string containing the FSI Viewer JS software version.	

strVersion	getBuild();
Returns a string containing the FSI Viewer JS software build number.	

void	showMenuToolTip(strTip, elSrc, x, y)
Shows custom tooltips.. strTip: HTML code of the tooltip or ID in language.xml files elSrc: the DOM element which is used to align the tooltip, usually event.target x,y: (optional) offset for tooltip position	

void	hideMenuToolTip()
	Shows custom tooltips.. strTip: HTML code of the tooltip or ID in language.xml files elSrc: the DOM element which is used to align the tooltip, usually event.target x,y: (optional) offset for tooltip position

oPageLayoutData	getPageLayoutData()
	oPageLayout has the following properties: boolean bDoublePages int nImages int nMaxPage int nMinPage

oParameters	getParameters();
	Returns an object containing all parameters set via javascript or custom tag attributes.

mixed	getConfigValue(strName);
	Returns the value of the configuration parameter "strName".

mixed	togglePageIndex();
	Toggles the page index.

bResult	makeBoolean(mixedConvert, bDefaultValue);
	Transforms a value into a boolean value, e.g. "1" to true, "false" to false <i>bDefaultValue</i> is optional and false if not defined.

mixed	getPluginParameter(strPluginName, strParameterName, defaultValue, bMakeBoolean);
	Returns the parameter defined for the given plugin. <i>defaultValue</i> and <i>bMakeBoolean</i> are optional. example: <code>iViewer.getPluginParameter("pages", "dir");</code>

7.4.2 Callbacks

You can define callback functions by assigning methods to the FSI Pages JS instance.

The following callback methods exist:

	onReady();
	Called as soon as FSI Pages JS finished loading data and gets interactive.

	onDestroy();
	Called when the fsi-pages element is destroyed.

	onChangeConfig(strCfgFileName, oParameters);
	Called when the config file is changed.

	onMenuEnabled(bEnabled);
	Called if the menu is enabled/disabled.

	onMenuDestroy();
	Called if the menu is destroyed.

onMenuButtonEnabled(strButtonID, null, bEnable);

This optional callback function can be used call an action when a button is enabled or disabled.

onMenuButtonPressed(strButtonID, evt);

This optional callback function can be used to set an action that starts when a button is pressed.

onMenuButtonReleased(strButtonID, evt);

This optional callback function can be used to set an action when a certain button is released..

onMenuButtonClicked(strButtonID, evt);

This optional callback function can be used to set an action when a certain button is released..

onMenuButtonSetPressed(bPressed);

This optional callback function can be used to call an action when a button is pressed.

onRegisterExternalMenuButtons()

This optional callback function can be used to call an action as soonexternal menu buttons are registered.

onAfterRegisterExternalButton(elButton, strButtonID, bButtonPresentInViewer)

This optional callback function can be used to remove inactive buttons from the menu bar if you are using a custom skin.

	onMenuReady();
	This optional callback function can be used to set an action when the menu is ready (after all buttons are added).

	onFullScreen(bSet);
	This optional callback function can be used to call an action when FullScreen mode is enabled or disabled.

	onPageChanged(nPage, strPage, strMaxPage);
	This optional callback function can be used to call an action as soon as the viewed page changes.

	onShowPageIndex(bShow);
	This optional callback function can be used to call an action as soon as the page index is opened.

	onShowBookmarkList(bShow);
	This optional callback function can be used to call an action as soon as the bookmark list is opened.

	onShowChapter(bShow);
	This optional callback function can be used to call an action as soon as the chapters opened.

	onShowPrintDialog(bShow);
	This optional callback function can be used to call an action as soon as the print dialog is opened.

onAutoPageLayoutChange(bDoublePage);

This optional callback function can be used to call an action as soon as the page layout is automatically changed.

onPageLayoutChange(oPageLayoutData);

This optional callback function can be used to call an action as soon as the page layout is manually changed.

onBookmarkAdded(nPage);

This optional callback function can be used to call an action as soon as a bookmark is added to the page.

onBookmarkRemoved(nPage);

This optional callback function can be used to call an action as soon as a bookmark is removed of the page.

onBookmarksLoaded(strLoaded);

This optional callback function can be used to call an action as soon as the bookmarks are loaded.

onBookmarksSaved(strLoaded);

This optional callback function can be used to call an action as soon as the bookmarks are saved.

onRemoveAllBookmarks();

This optional callback function can be used to call an action as soon as all bookmarks are removed.

	onPageSizeChanged(nWidth, nHeight);
This optional callback function can be used to call an action as soon as the page size is changed.	

	onPageZoomed(nPage);
This optional callback function can be used to call an action as soon as the page size is zoomed.	

	onPrintPage(arPageIndices, arImageURLs);
This optional callback function can be used to call an action as soon as the pages are printed.	

	onPageToolTip(nLinkAreaIndex, strURL, strTipContent);
This optional callback function returns a string that is useful if you want to modify the tool tip content.	

	onOpenPageLink(nLinkAreaIndex, strURL, strTarget);
This optional callback function can be used to call an action if links are clicked. Return "false" to prevent opening the link., return {strURL:"foo", strTarget: "bar"} to modify the URL or the target (both are optional)	

	onResize(nWidth, nHeight, bResizeDone, BFullScreen);
Called when the fsi-pages element is resized.	

	onSkinChanged(strCurrentSkinClass, strCurrentSkinClassBefore)
This optional callback function can be used to call an action as soon as the skin class is changed.	

8 FSI Viewer

8.1 Compatibility and User System Requirements

Imaging Servers

FSI products covered by this manual can at present securely be run with FSI Server from version 2016. FSI Viewer 6.0.0 does no longer support eRez Imaging Server, iSeeMedia Zoom Image Server and TrueSpectra.

Adobe Flash™

FSI Viewer, FSI Pages Add-on and FSI Showcase Add-on require Flash™ Plug-in version 6 or newer to be displayed in browsers.

FSI Viewer has been successfully tested with the following software versions:

FSI Server

Build 16.07 or newer

Adobe Flash™ Plug-in

6.0.65.0 to 16.0.0.305

8.2 Using FSI Viewer

FSI Viewer is navigated by the menu bar (here seen below the image) and by using the mouse directly on the image. For example, you can select a section of the image you want to magnify. The mouse functions are determined by the corresponding buttons on the menu bar (zoom, pan, rotate, etc.). The example to the right shows the mouse mode "zoom".






The optional small navigator window (bottomright) displays the position of the image section currently viewed.

8.3 The Menu Bar







The menu bar might look different depending on skins or additional plugins.

Main Functions



	<p>Back to Initial View. Undoes zoom, pan and rotation and restores the initial view. (identical with pressing space bar)</p>
	<p>Zoom In</p>
	<p>Zoom Out</p>

8.4 Mouse Modes

The `→ Mousemodes` or `→ MousemodeSelect` plug-in is required to display the following mouse mode buttons.

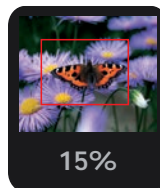
	<p>Mouse - Zoom In Choose this function to enlarge segments using the mouse. Click on the image and drag the frame over the desired segment. Alternatively you can click on the image, without marking a segment. The image will then be magnified in steps. To zoom out in steps, hold down the CTRL-key and click on the image.</p>
	<p>Mouse - Pan Choose this function to pan the image using the mouse. Click on the image and drag in the desired direction. To return to the original view, hold down the CTRL-key and click on the image.</p>
	<p>Mouse - Turn Choose this function to rotate the image around the z-axis. Click on the image and drag in whatever direction you want to turn the object (to the right or the left). To reset the rotation, hold down the CTRL-key and click on the image.</p>
	<p>Mouse - Rotate (only for 3D presentations) Choose this function to rotate the object around the y-axis or the x-axis if available. Click on the image and drag to the left or to the right. Move the mouse up or down to rotate the object around the x-axis. To reset rotation, hold down the CTRL-key and click on the image.</p>

8.5 Additional Buttons

	<p>Hide / Display menu Displays or hides the user interface.</p>
	<p>Information Displays information about the viewer and refers to a configurable help page (Parameter → <i>HelpURL</i>).</p>

8.6 Using the navigator window

In the navigator window you can see a miniature presentation of the entire image. The segment which is currently viewed is framed. In the navigator window you can change the current segment either by dragging the frame or by clicking on the desired area of the image.



8.7 Keyboard Shortcuts

FSI Viewer, FSI Pages and FSI Showcase can additionally be controlled using the keyboard. The list below shows the key function assignments.

Key	Function
Num 5, spacebar	Back to original view
Num 4, left arrow	Move to the left
Num 6, right arrow	Move to the right
Num 8, up arrow	Move up
Num 2, down arrow	Move down
Num 1	Turn around the z-axis to the left
Num 3	Turn around the z-axis to the right
Num 7	Rotate around the y-axis to the left (3D only)
Num 9	Rotate around the y-axis to the right (3D only)
Num /	Rotate around the x-axis (3D only) or previous view
Num *	Rotate around the x-axis (3D only) or previous view
Num 0	Hide/display user interface
D	If you have activated the debug mode, you can hide or display the debug window using this key.
I	Refresh the image information in the debug window.
Pos 1	FSI Pages: First Page
Page Up	FSI Pages: Previous Page
Page Down	FSI Pages: Next Page
End	FSI Pages: Last Page

8.8 Index structure of FSI Viewer

FSI Viewer, FSI Pages and FSI Showcase can additionally be controlled using the keyboard. The list below shows the key function assignments.

Directory	Content Description
core	Core components
debug	Debugging components <i>This folder can safely be deleted if debugging is not required any more</i>
languages	User interface language files
plugins	FSI Plug-ins extending FSI Viewer at runtime
skins	FSI Skins containing the visible parts of the user interface

Please Note: You need to make sure that your web server properly serves files with the following extensions: *.swf, *.xml

8.8.1 Licensing FSI Viewer

You can use FSI Viewer without a licence key for evaluation purposes. In this case the phrase "FSI Viewer Evaluation Copy" appears on top of the image and the about box displays "Unregistered Evaluation Copy" in the "Licenced to" section. After purchasing an FSI Viewer licence you will receive a licence key that removes the "FSI Viewer Evaluation Copy" mark and the name of the licensee appears in the "licenced to" section of the about box.

Installing the Licence Key

The FSI Viewer Licence Key needs to be installed together with the FSI Server Licence Key. For more information, please refer to the relevant section in the FSI Server manual.

Please note that FSI Viewer will run in evaluation mode regardless of a valid licence key if you install FSI Server on a different machine than you have licensed it to. The same applies to unregistered FSI Plug-ins or FSI Skins. In this case please check the FSI Viewer debug window (→ *Using the Debug Window*, which provides information on the licensing status and contact us if the problem persists.

8.9 Integration into HTML Pages

8.9.1 Required HTML-Source Code

FSI Viewer can be integrated into HTML pages by means of an HTML `<object>` tag - just like any other Adobe Flash movie clip. Configuration parameters can be appended to the URL of the movie clip in the form of HTTP queries. If you already know how to add Flash movie clips to HTML code you can skip this chapter and move on to the chapter → *Configuration*.

Object Tag

All recent browsers require an `<Object>` tag to integrate a Flash movie clip. Older browsers use the obsolete `<Embed>` tag instead. Sample HTML code to integrate FSI Viewer into HTML pages: (Variable data printed in bold and square brackets):

HTML integrations for modern and old browsers

```
<Object classid="clsid:D27CDB6E-AE6D-11cf-96B8-
444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cab
s/
flash/swflash.cab#version=6,0,65,0"
width="[Width]" height="[Height]">
<PARAM name="movie" VALUE="[URL and Parameter]" />
<PARAM name="bgcolor" VALUE="[Background color]" />
<PARAM name="menu" VALUE="false" />
<PARAM name="allowscriptaccess" value="always" />
<PARAM name="allowfullscreen" value="true" />
<PARAM name="quality" value="high" />
</Object>
```

Embed Tag

For browsers (e.g. Netscape 4.x) that do not support the `<object>`-tag, an additional, encapsulated `<embed>`-tag with identical parameters is required.

HTML integrations for modern and old browsers

```
<Object classid="clsid:D27CDB6E-AE6D-11cf-96B8-
444553540000"
codebase="http://download.macromedia.com/pub/shockwave/
cabs/flash/swflash.cab#version=6,0,65,0" width=" [Width] "
height=" [Height] ">
<PARAM name="movie" VALUE="[URL and Parameter]" />
<PARAM name="bgcolor" VALUE="[Background color]" />
<PARAM name="menu" VALUE="false" />
<PARAM name="allowscriptaccess" value="always" />
<PARAM name="allowfullscreen" value="true" />
<PARAM name="quality" value="high" />
<EMBED name="application/x-shockwave-flash"
PLUGINSPPAGE="http://
www.macromedia.com/shockwave/download/index.cgi?P1_Prod_
Version=ShockwaveFlash" SRC="[URL and Parameter]" WIDTH="
[Width]"
HEIGHT="[Height]" BGCOLOR="[Background color]" MENU="false"
ALLOWSCRIPTACCESS="always" ALLOWFULLSCREEN="true"
QUALITY="high"></
EMBED>
</Object>
```

Variables in the <object> and <embed> tags:

[Width]	Width of the viewer in pixel or percent
[Height]	Height of the viewer in pixel or percent
[Background color]	Background color, hexadecimal (e.g. #FFFFFF for white)
[URL and parameter]	URL to FSI Viewer and parameters provided as HTTP query, e.g. "?cfg=foo"

For further possible parameters or XHTML configuration options, please refer to Flash documentation or renowned HTML references.

8.10 Configuration

Configuration parameters can be provided using XML- configuration files and/or by HTTP queries appended to the URL of the Flash movie clip.

Providing parameters by HTTP query is recommended if you plan to display large numbers of images (→ *Automated Implementation of Images*).

In the simplest case you only define the relative path to a configuration file within the "src" parameter of the <object>-Tag using the "cfg" parameter:

Defining the CFG parameter inside the object tag

```
<PARAM name="movie" VALUE="fsi.swf?cfg=[relative path to FSI-  
file]">
```

The following table lists FSI parameter value types, referred to in this documentation:

Type	Example
Number	"90"
String	"ZoomIn"
URL	"http://www.fsi-viewer.com/"
Boolean	"0" / "1" or alternatively "true" / "false"
HexColor	"FF0000"

8.10.1 Using XML Configuration Files (*.xml)

FSI Viewer uses XML configuration files which can easily be edited with any text editor.

The default configuration file "_default.xml" - which is located in the same folder as the main viewer file "fsi.swf" - is the most important XML configuration file and will always be evaluated first.

To use FSI Viewer with additional configuration files you have to provide the name of the configuration file using the "cfg" parameter attached to the URL of the "fsi.swf" file:

Specifying an XML configuration file for FSI Viewer

```
PARAM name="movie" VALUE="/fsi/fsi.swf?cfg=foo">
```

Configuration files are structured into configuration groups (XML nodes) containing parameter names and values. Only parameters in these groups will

be recognized by FSI Viewer. Configuration files do not need to contain all groups though. Each parameter has to be provided as an individual XML child node of a group where the node name is the parameter name and the parameter value needs to be provided as "value" attribute of the node.

Structure of XML configuration files

```
<fsi_parameter>
  <image>
    <parameter value="" />
  </image>

  <plugins>
    <parameter value="" />
  </plugins>

  <options>
    <parameter value="" />
  </options>
</fsi_parameter>
```

Simple XML configuration file

```
<fsi_parameter>
  <image>
    <path value="samples/Watch.jpg" />
  </image>
  <options>
    <skin value="silver" /><
    debug value="1" />
  </options>
</fsi_parameter>
```

Please keep in mind the following rules when using XML configuration files:

- attribute values must be provided in double quotes
- the last occurrence of a given parameter will be used
- configuration files might be cached by your browser, so that changes in these files require flushing the browser cache. This does not apply when using the debug mode.
- make sure to save configuration files in UTF-8 format if your configuration file contains language specific diacritics, Japanese characters and alike

You can comment on or disable sections of an XML configuration file using the usual XML comment syntax:

Comments and disabling parameters temporarily

```
<image>

    <!-- This is a comment -->
    <path value="image.tif" />

    <!-- The following section will be ignored -->
    <!--
    <path value="anotherimage.tif" />
    -->

</image>
```

8.10.2 Using HTTP Queries

In addition to using XML configuration files you can define FSI parameters by adding a HTTP query to the URL of the FSI Viewer movie clip. In this case parameters have to be provided in the following form:

Parameter1=Value1&Parameter2=Value2...



Characters that are not URL compatible such as "/", "% "and "& "within the values need to be "URL-encoded.

Due to the missing possibility to define parameter groups when providing FSI parameters by HTTP query you have to prefix parameters with the name of the parameter group they belong to.

Prefixing parameters with group names

```
XML Syntax:

<image>
    <path value="foo.tif" />
    <servertime value="fsi" />
</image>

Equivalent HTTP Query:
```

Prefixing parameters with group names

```
?imagepath=foo.tif&imageservertype=fsi
```

Most parameters belong to the global "<options>" group. These parameters do not need to be prefix in order to keep HTTP queries as short as possible.

Providing <Options> parameters by query

XML Syntax:

```
<options>
  <debug value="true" />
  <menualign value="TL" />
</options>
```

HTTP Query equivalent:

```
?debug=true&menualign=TL
```

The FSI parameter reference provided in this documentation contains the parameter names as well as the context (name of the parameter group) they belong to.

8.10.3 Hierarchy of Configuration Parameters

As explained before parameters can be provided in three different ways:

1. The default configuration file "_default.xml" - located in the same folder as the main viewer file "fsi.swf" - will always be evaluated first
2. You can store image specific parameters in additional *.xml files
3. You can provide parameters by HTTP query

Parameters specified in multiple ways – for example in the default configuration and by HTTP query – will be evaluated in the following way:

- Parameters defined in configuration files override parameters defined in the default configuration file "_default.xml".
- Parameters provided by HTTP query override parameters in all configuration files.

The list below shows the hierarchy resulting from the sequential evaluation of the different configuration methods:

Configuration Hierarchy

1. HTTP Query
2. Configuration file (*.xml)
3. Default configuration file (_default.xml)

Choosing the adequate Configuration Method

So far you learned that FSI Viewer can be configured by any combination of:

- The default configuration file (_default.xml)
- Additional configuration files (*.xml)
- HTTP queries

These configuration options are not only alternate methods. Combining these methods offers the possibility to provide configuration parameters as easily as possible and as dynamically as required. The following considerations might be helpful when deciding which method to choose for a certain parameter.

8.10.4 Default Configuration

You can easily change global parameters that apply to all FSI Viewer instances by editing a single file. This is therefore the recommended configuration method for:

- Global parameters valid for all images, in particular:
 - ImageServerType
 - Licence Key
- General configuration options like:
 - Skin
 - Intro
 - Language
 - Plug-ins
 - MenuAlign, AnimationSpeed and alike

8.10.5 Configuration Files

Separate configuration files should be used for parameters specific to a group of images or a specific way to display images. You might for example want to setup individual configuration files for different types of FSI Viewer instances, like "small_no_interface.xml", "large_with_hotspots.xml" as an example.

Try to avoid adding global parameters like the base path to your imaging server "ImageServer" or alike in individual configuration files. This way you will only

need to change the "_default.xml" configuration file once you need to change the address of your imaging server.

Usually you don't need to create a configuration file for individual images, as you can specify the image path or other unique parameters dynamically by HTTP query.

8.10.6 Passing Parameters by HTTP Query

This method is recommended for parameters from databases, parameters specific to a certain image or FSI Viewer instance, or parameters related to user interaction. Using server side script or JavaScript you can build HTTP queries to pass parameters to FSI Viewer dynamically.

If you want to display multiple images with minimal variations in the FSI Viewer configuration, it is recommended to reuse the same FSI configuration file and to provide varying parameters (like the path to the image itself) by HTTP query.

Try to provide only those parameters by HTTP query that don not apply to multiple instances of FSI Viewer. Keep in mind that changing a single configuration file for multiple images later on will be far less work compared to editing multiple HTML codes later on.

Providing parameters by HTTP query

View 1:

```
<PARAM name="movie"  
VALUE="fsi.swf?cfg=flowers&imagepath=rose.fpx">
```

View 2:

```
<PARAM name="movie"  
VALUE="fsi.swf?cfg=flower&imagepath=rose.  
fpx&NoNav=1">
```

8.10.7 Retrieving Parameters from Imaging Servers

FSI Viewer is capable of retrieving image specific parameters from FSI Server dynamically. This way you do not need to define these parameters explicitly. The list below shows some typical parameters that can be retrieved from FSI Server automatically:

- **ImageWidth**

- **ImageHeight**
- **ImageTilesX**
- **ImageTilesY**
- **SceneSets**
- **Image Collections**
- **Hyperlinks**
- **Hotspots**

In which case will these values be retrieved automatically?

FSI Viewer tries to retrieve the parameters ImageWidth and ImageHeight from the server if one of these values has not been defined manually. If the server delivers additional parameters (e.g. ImageTilesX, ImageTilesY) these parameters will only have an effect if they have not been defined manually.

8.10.8 Parameters

Basic Parameters

The following parameters are essential and usually need to be specified. Please refer to the chapter "Retrieving Parameters Automatically" for information on how to enable FSI Viewer to retrieve image specific parameter values from the imaging server.

cfg	
Description	Relative path to a configuration file
Syntax	fsi.swf?cfg=[FSI_Name]
Default	—
Context	HTTP query

A relative path to an XML based configuration file `"*.xml"`. The file extension `".xml"` has to be omitted. You can specify a default base path to all configuration files using the `"FSIBase"` parameter.

Defining a configuration file

```
<PARAM name="movie" VALUE="fsi.swf?cfg=foobar">
```

In this case the FSI Viewer first searches for the `_default.xml` file in the installation folder where the file `fsi.swf` is located and evaluates the configuration parameters. The optional parameter `→ FSIBase` from the

default.xml file is then used as the path specification in order to search in the correct index for a FSI configuration file with the name "foobar.xml".

DefaultCfg	
Description	Relative path to an alternate default configuration file
Syntax	fsi.swf?defaultcfg=[filename]
Default	_default
Context	HTTP query only

Relative path to an alternative default configuration file other than "_default.xml". Only useful as part of an HTTP query. The file extension ".xml" has to be omitted.

server	
Description	URL to your FSI Server
Syntax	URL
Default	See description below
Context	<Image>
Version	5.5.0 or higher

The URL to your FSI Server including the servlet name , e .g. :

```
<ImageServer value="http://mydomain.tld/fsi/server" />
```

If you do not specify this parameter, FSI Viewer will try to determine the URL based on it's own URL and the default location of the „viewer" folder in the FSI Server setup.

ServerTemplate	
Description	Real-time template of the imaging server
Syntax	String (template name)
Default	fsi
Context	<Image>
Query	ImageServerTemplate
Version	5.5.0 or higher

This parameter specifies the name of a real-time template used when accessing the imaging server. Using these templates you can specify e.g. the image compression, maximum image dimensions and watermarks.

path	
Description	Path of the image to display
Syntax	String
Default	—
Context	<Image>
Version	5.5.0 or higher

The path to the image on your FSI Server including the importer profile name, e.g.:

```
<ImagePath value="images/samples/foo.jpg" />
```

width and height	
Description	Source image dimensions
Syntax	Number in pixels
Default	—
Context	<Image>
Query	ImageWidth and ImageHeight
Version	5.5.0 or higher

Required parameters specifying the width and height of the source image. Please refer to the chapter → *8.10.7 Retrieving Parameters from Imaging Servers* for information on how to automatically retrieve these values from the imaging server.

Width and height specifications for a source image

```
<image>
...
    <Width value="3330" />
    <Height value="4660" />
...
</image>
```

Optional Parameters

The following parameters are not required to run FSI Viewer. They can be used to alter the appearance of the FSI Viewer user interface.

skin	
Description	Filename of FSI Skin to use (without extension)
Syntax	String
Default	—
Context	<Options>

Filename of FSI Skin file to use (omit the file extension). FSI Skins determine the appearance of the user interface and are located in the subdirectory "/skins". In addition to the default skins that ship with FSI Viewer customized skins are available on demand. Please note that using unregistered skins activates the evaluation mode.

intro	
Description	Filename of start animation to use (without extension)
Syntax	String
Default	—
Context	<Options>

Filename of animation to display on startup (omit the file extension). FSI Viewer ships with a default start animation. Custom intros are available on demand. Please note that using an unregistered intro activates the evaluation mode.

debug	
Description	Activate debug mode
Syntax	Boolean
Default	false
Context	<Options>

Activate the debug mode providing configuration details, warnings and error messages.

It is strongly recommended to activate the debug mode when encountering any issues running FSI Viewer.

Please refer to the chapter → *8.12 Debug Mode* for a detailed description.

FSIBase	
Description	Path to FSI configuration files
Syntax	Relative path
Default	config/
Context	<Options>

Enter the relative path from FSI Viewer to the image specific configuration files here. If this parameter is specified the path is added to all relative CFG parameters.

Definition of FSIBase in the _default.xml file

```
<options>
...
    <FSIBase value="config/" />
...
</options>
```

Definition of FSIBase in the _default.xml file

```
<object ...>
...
    <PARAM name="movie" VALUE="/fsi/fsi.swf?
    cfg=viewer_presets/no_nav">
...
</object>
```

The content of the parameter "CFG" will be prefixed by the parameter "FSIBase" to "config/viewer_presets/no_nav.xml"

CropRect	
Description	Defines a crop rectangle
Default	0,0,1,1
Context	<Options>
Version	5.0.0

Crop rectangle (left, top, right, bottom) as 0.0..1.0 floating point values to display a given section of an image only.
Default: "0,0,1,1"

Note:

You cannot use "CropRect" AND "CropValues" parameters at a time.
If you use both, "CropRect" will be ignored.

CropValues	
Description	Defines crop margin values for an image
Syntax	String
Default	0,0,0,0
Context	<Options>
Version	5.6.0

Defines the amount that should be cut off the image from each side.
Values can be provided as pixel (e.g. "20px") or percent (e.g. "10%") or floating point numbers (e.g. "0.25").

MenuAlign	
Description	Alignment of the menu bar
Syntax	String
Default	version 1-4: TL, from version 5: BC
Context	<Options>
Version	The "center" option "C" is available from version 5.0.0

Possible values:

TL	(top-left)	TC	(top-center)
TR	(top-right)	BL	(bottom-left)
BC	(bottom-center)	BR	(bottom-right)

language	
Description	Interface language
Syntax	String
Default	---
Context	<Options>
Version	FSI Viewer 2.0 or higher / FSI Showcase 2.15 or above

Defines the language of tool tips for the user interface.

Possible values depend on the XML based language files located in the sub-index

"/languages" of the FSI Viewer setup path. To specify a language, please enter the

filename without file extension, e.g. "german".

You can modify existing or add your own "language files".

ImagePanButtons	
Description	Show or hide pan buttons on the image
Syntax	Boolean
Default	false
Context	<Options>
Version	4.0.0 and above

Show or hide the optional pan buttons on the image. If the user points at a pan button, the image pans into the direction indicated. The buttons automatically disappear if the edge of the image has been reached.

DetailBuffer	
Description	Enable or disable the detail buffer
Syntax	Boolean
Default	true
Context	<Options>
Version	3.0.0 and above

Enabling the detail buffer keeps previously loaded image details visible when zooming in or out instead of displaying the low resolution preview image while loading additional image details. This way zoom transitions appear smoother.

NoNav	
Description	Hide the navigator window
Syntax	Boolean
Default	false
Context	<Options>

Show or hide the navigator window displaying the miniature image and the current image section.

AutoHideNav	
Description	Hide navigator window if magnification is lower or equal to 100%
Syntax	Boolean
Default	true
Context	<Options>
Version	6

Show or hide the navigator window displaying the miniature image and the current image section, if the magnification is lower or equal to 100%.

NavWidth and NavHeight	
Description	Maximum size of the navigator window
Syntax	Number in percent or pixel
Default	80
Context	<Options>
Version	3.0.0 and above

Defines the maximum width and height of the image inside the navigator window. The actual size will depend on the image aspect ratio. The value can be defined absolutely or in percent of the viewer size. Use parameter "NoNav" to hide the navigator window.

NavFrameColor	
Description	Color of the selection frame inside the navigator window
Syntax	HexColor
Default	FF0000 (depending on the skin)
Context	<Options>
Version	3.0.0 and above

Specify a 6-digit HexColor to change the color of the selection frame inside the navigator window, e.g. "0000FF" for blue.

NavMaskColor	
Description	Color of the area around the selection frame of the navigator window
Syntax	HexColor
Default	depends on the skin
Context	<Options>
Version	3.0.0 and above

Specify a 6-digit hexadecimal color to change the color of the area outside the selection frame of the navigator window, e.g. "FFFFFF" for white.

NavMaskAlpha	
Description	Opacity of the area around the selection frame of the navigator window
Syntax	Number (0..100)
Default	depends on the skin
Context	<Options>
Version	3.0.0 and above

Specify the opacity of the area outside the selection frame of the navigator window.

Possible values are integer values from "0" (invisible) to "100" (opaque).

InitialView	
Description	Initial image segment
Syntax	SceneSet, Scene [left, top ,right, bottom, rotation]
Default	1,1,0,0,1,1,0
Context	<Options>

Image section (and rotation) to display on startup.

The first two parameters are required as they specify the scene set and the scene. For 2D images both values default to 1. The other parameters are optional and specify the image segment and the rotation around the z-axis. Please use the plug-in → *SelectFrame* to easily retrieve valid values for this parameter.

InitialView
<pre><Options> ... <InitialView value="1,5,0,0,0.25,0.25,90" /> ... </Options></pre>

The example displays the top left quarter of the 5th scene in the 1st scene set, rotated 90° to the right.

InitialViewPersistent	
Description	Keep the InitialView as default view
Syntax	Boolean
Default	false
Context	<Options>
Version	2.0.0 or higher

If this parameter is activated, clicking the "Reset" button will display the image section defined by the InitialView parameter instead of the entire image. If this parameter is set to "false", the initial view will be displayed on startup, but pressing the Reset button will reset the view to the entire image.

InitialAction	
Description	Action on startup
Syntax	String
Default	—
Context	<Options>

Specifies an action to execute on startup. The action is repeated until the user presses a button or key.

While the action is being executed no image details are loaded.

Possible values:

NextScene	(rotate around the y-axis to the right, 3D only)
PreviousScene	(rotate around the y-axis to the left, 3D only)
NextSceneSet	(rotate around the x-axis to the right 3D only),
PreviousSceneSet	(rotate around the x-axis to the left, 3D only)
RotateRight	(rotate around the z-axis to the right, 3D only)
RotateLeft	(rotate around the z-axis to the left, 3D only)
ZoomIn	(zoom in)
ZoomOut	(zoom out)

InitialActionDelay	
Description	Speed of the initial action
Syntax	Number
Default	3
Context	<Options>

Specifies the speed of the action specified by the InitialAction parameter. Possible values are integer values greater than 0.

0 highest speed
>1 slower

InitialActionPersistent	
Description	Repetition of initial action
Syntax	Boolean
Default	false
Context	<Options>
Version	2.2.0 or higher

Initial action restarts on each 'reset' command (button or keyboard) if set to true. Otherwise the action only performs on startup.

HelpURL	
Description	URL of custom help page
Syntax	String
Default	http://help.fsi-viewer.com
Context	<Options>
Version	1.3.0 or higher

If you want to create a custom help page for your FSI Viewer enter the complete (absolute) URL to this page using this parameter.

HelpURLTarget	
Description	Target (frame) for custom help page
Syntax	String
Default	_blank
Context	<Options>
Version	1.3.0 or higher

Using this parameter you can define the target name of the browser window if a user opens the help page. The default value "_blank" opens a new browser window.

Initial MouseMode	
Description	Mouse mode on startup
Syntax	Number
Default	0
Context	<Options>

Specifies the selected mouse mode on startup.

Possible values:

- 0: (Zoom)
- 1: (Pan)
- 2: (Rotate 3D)
- 3: (Rotate 2D)

Some plug-ins provide additional mouse modes. Please refer to the corresponding plug-in reference → *Mousemodes* for details.

EnableZoomInPanMode	
Description	Enable zooming in pan mode
Syntax	Boolean
Default	false
Context	<Options>
Version	4.1.2

Set this parameter to true if you want to enable the user to zoom by clicking the image while mouse mode "pan" is enabled.

EnableResetInPanMode	
Description	Reset zoom when clicking on the image at max magnification
Syntax	Boolean
Default	true
Context	<Options>
Version	6

Set this parameter to true if you want to reset zoom when clicking on the image at maximum magnification. This parameter has no effect if "EnableZoomInPanMode" is enabled.

HandCursor	
Description	Shape of the mouse cursor
Syntax	Boolean
Default	true
Context	<Options>

If you set this parameter to "false" the default arrow cursor will be used instead of the hand cursor when hovering over the image or the navigator window.

ZoomLimit	
Description	Maximum magnification
Syntax	Number
Default	100
Context	<Options>

Defines the maximum magnification in percent relative to the source image. The default value "100" permits magnification to the physical resolution of the source image. Possible values are "1" to "1000".

ZoomLimitMin	
Description	Minimum magnification
Syntax	Number
Default	100
Context	<Options>

Defines the minimum magnification in percent relative to the initial image. The default value "100" permits magnification to the size of the Viewer. Smaller values permit magnifications smaller than the viewer size. Possible values are "1" to "100".

FlickMotion	
Description	Enables dragging lists in addition to using the scroll bar
Syntax	Boolean
Default	true
Context	<Options>
Version	6

With this option enabled it is possible to drag lists in addition to using the scroll bar. This can be used for e.g. FSI Showcase thumbnails, FSI Pages index, etc.

FlickMotionOverDrag	
Description	allows dragging the content slightly over min/max
Syntax	Boolean
Default	true
Context	<Options>
Version	6

Allows dragging the content slightly over the minimum and maximum and the content bounces back to the limit. Only applicable if "FlickMotion" is enabled.

ExtendedViewport	
Description	Extend the size of the viewport
Syntax	Boolean
Default	false
Context	<Options>

With this option enabled you can pan the borders of the image to the center of the view port. If you disable this option the panning range is smaller so that the user can not pan the image borders to the center of the view port.

StrictViewport	
Description	Strict viewport allowing minimum image panning only
Syntax	Boolean
Default	false
Context	<Options>
Version	4.1.0 or above

With this option enabled the pan range for the image is limited to the minimum required to view the entire image. This option can be useful if you want the user to pan in one direction only, because the image width matches the view port width or height (e.g. when using FSI Pages with "autozoom=fitWidth" option).

MagnifySmallImages	
Description	Magnify images smaller than the viewport
Syntax	boolean
Default	false
Context	<Options>
Version	4.1.0 or above

Usually FSI Viewer 4.1.0 or above does not magnify small images to fit the view port.

You can enable this option in case you want small images to fit the view port even if the source image is smaller than the view port.

Animation	
Description	Animation quality
Syntax	Boolean
Default	true
Context	<Options>

Animate zoom and rotate actions (true) or zoom and rotate in a single step (false).

AnimationType	
Description	Defines type of zoom animation
Syntax	Number
Default	3
Context	<Options>
Version	5.0.0

Defines the type of the zoom animation (acceleration on start or end, bounce, etc.)

Possible values are 0 to 6.

AnimationSpeed	
Description	Animation speed
Syntax	Number
Default	50
Context	<Options>

Specifies the speed of the zoom and scene animations. Possible values range from 1 (very slow) to 100 (very fast). See parameter "Animation" to disable animated motion.

HideUI	
Description	Hide user interface
Syntax	Boolean
Default	false
Context	<Options>

Hide the menu bar on startup. The user has to click the "Show Menu" button to show the user interface. This option is especially useful when an image is meant to look like an ordinary image or if there are many FSI Viewers on a single web page.

RoundedSkinCorners	
Description	Defines round skin corners
Default	—
Context	<Options>
Version	5.0.0

Defines the radius of the rounded viewer corners. Possible values range from 0 (no rounded corners) to 99.

UISwitchable	
Description	Enable hiding/showing the user interface
Syntax	Boolean
Default	true
Context	<Options>
Version	3.2.2 and above

Specifies whether the user is able to show or hide the user interface by clicking the corresponding button or pressing "0" on the num pad. The "Show Menu" button is hidden if this option is set to "false".

ViewerBorder	
Description	Defines the color of the viewer borders
Default	false
Context	<Options>
Version	5.0.0

Color of the border around the viewer instance as a hexadecimal color value, e.g. FF0000 for red. Use "false" for no border.

NoImageBlend	
Description	Subsequently loaded image details display without transition
Syntax	Boolean
Default	false
Context	<Options>
Version	1.2.0 or higher

If this parameter is activated, subsequently loaded image details are displayed abruptly without transition.

MobileWindowTitle	
Description	Defines the browser title text for mobile devices
Syntax	String
Default	(As defined in index.html)
Context	<Options>
Version	5.6.0

Specifies the window browser title on mobile devices for e.g. FSI Pages mobile.

AutoElevateToHTTPS	
Description	switches protocol to https
Syntax	Boolean
Default	true
Context	<Options>
Version	5.7.0

Automatically switches the protocol used by the different Viewers to https if required by the setup.

Effects	
Description:	Image manipulation parameters for all images
Syntax:	String
Default:	—
Context:	<Options>
Version:	1.3.0 or higher

The effects parameter can be used to pass image manipulation parameters to the imaging server. Typical manipulation parameters include the image saturation ("saturation") and image sharpening ("sharpen"). If no effect is specified, the default settings of the server (profile) apply. Please refer to your FSI Server documentation for a list of available effect parameters.

Example "temperature"

The "temperature" parameter changes the color temperature of images delivered by the imaging server.



Practical values range from -100 to max. 100 Example – temperature(-100) <effects

Example – temperature(100)

```
<effects value="temperature(100)" />
```

Example "pixelize"

The "pixelize" parameter or sharpens images creates a pixel effect by reducing the image details of the images delivered by the imaging server.

Example –pixelize(50)



```
<effects value="pixelize(50)" />
```

Other effects change, for example, the color saturation, sharpness, hue or contrast. Please refer to your imaging server documentation for a complete description of available image effect commands.

Example –brightness(10)



```
<effects value="brightness(10)" />
```

Example –brightness(10)

```
<effects value="brightness(-10)" />
```

Multiple effects can be applied by concatenating the effect parameters by commas.

Example:

```
<effects value="Brightness(12),Sharpen(40)" />
```

Please note that when passing effect parameters by HTTP query, the value needs to be provided as an url-encoded string.

For example:

```
...fsi.swf?effects=Brightness%2812%29%2CSharpen%2840%29
```

8.10.9 Applying Effects to Specific Images Only

While the effects parameter holds image manipulation parameters for all images you can also apply effects to specific types of images in FSI Viewer (see → *NavEffects*, → *SceneEffects* and → *TileEffects*).

For example it is sometimes useful to sharpen the images inside the navigator window as small images tend to appear a little blurry. To do so you can add a corresponding "NavEffects" parameter.

Sharpening Images in the navigator window:

```
<NavEffects value="sharpen(35)" />
```

NavEffects	
Description	Image manipulation parameters applied to the navigator window
Syntax	String
Default	—
Context	<Options>
Version	3.1.1 or higher

Applies image manipulation effects like "sharpen" or "quality" to all images inside the navigator window. Please refer to "effects" parameter for a detailed description.

SceneEffects	
Description	Image manipulation parameters applied to preview images
Syntax	String
Default	—
Context	<Options>
Version	3.1.1 or higher

Applies image manipulation effects like "sharpen" or "quality" to all un-tiled images in the viewer. Please refer to "effects" parameter for a detailed description.

TileEffects	
Description	Image manipulation parameters applied to image tiles
Syntax	String
Default	—
Context	<Options>
Version	3.1.1 or higher

Applies image manipulation effects like "sharpen" or "quality" to all image tiles in the main view area. Please refer to "effects" parameter for a detailed description.

IgnoreQuery	
Description	Ignore all parameters passed via query (except cfg)
Syntax	Boolean
Default	false
Context	<Options>

Ignore all parameters passed via HTTP-query except for the "cfg" parameter. This way you can entirely disable modifications to the FSI configuration by query.

Obsolete parameters for ZoomCache software:

The successor FSI Cache does not require any special parameters.

IgnoreQueryParameters	
Description	List of parameters to ignore in queries
Syntax	String
Default	—
Context	<Options>
Version	5.0.4 or higher

A comma separated list of parameter names that should be ignored if specified in the HTTP query.

Example: <IgnoreQueryParameters value="language, skin" /> would prevent specifying the "skin" and "language" by HTTP query.

Please note:

- "DefaultCfg", "Cfg" and "Debug" parameters cannot be disabled
- This parameter itself cannot be specified by query

ZoomCache	
Description	URL of the ZoomCache
Syntax	URL
Default	—
Context	<Options>

If you use NeptuneLabs ZoomCache, please specify the URL of the ZoomCache server here. You can receive this URL and a ZoomCache ID from your image hosting provider.

ZoomCacheID	
Description	ID of the ZoomCache access profile
Syntax	String
Default	—
Context	<Options>

When using NeptuneLabs ZoomCache enter the name of the access profile for your images here. You may have several profiles i.e. if you use watermarking or different imaging servers.

MaxZoomLevel	
Description	Load image details at maximum zoom
Syntax	Boolean
Default	true
Context	<Options>
Version	3.2.1 or higher

By default FSI Viewer adds a zoom level for the maximum magnification – this is FSI Viewer loads image details when reaching the maximum magnification. For extremely high resolution 2D images exceeding 32,000 pixels in width or height

you can set this value to "false" to prevent display problems at the maximum magnification.

TiltModeRelative	
Description	Use absolute or relative tilt mode
Syntax	Boolean
Default	true
Context	<Options>
Version	3.6.0 or higher

True: Rotate the image (2D) around the center of the view port depending on the mouse position and distance to the center. False: Rotate the image (2D) clockwise moving the mouse right and counterclockwise moving the mouse left.

TileSizeX and TileSizeY	
Description	Size of image detail tiles
Syntax	Number in pixel or in percent of the viewer size
Default	50%
Context	<Options>
Version	2.3.0 or higher

Specifies the size of image details tiles. The value can be defined relative to the viewer size (e.g. "25%") or using absolute pixel values (e.g. "256"). The default value of 50% means that each image tile is one quarter in size of the entire viewer. This way a maximum of 9 image tiles is required to display the selected image section.

Please do not change this value unless you are certain that this is necessary and you are aware of possible consequences:

- Defining smaller image tiles reduces the overall viewer performance and increases the number of requests and thus the imaging server load.
- Defining larger image tiles might lead to increased traffic and decreased download performance as more image data than actually required for a certain image section will be loaded.

InfoTemplate	
Description	Configuration template to retrieve image info from server
Syntax	String
Default	fsi/info.xml
Context	<Image>
Query	ImageInfoTemplate
Version	5.5.0 or higher

Specifies the configuration template used to retrieve image specific data from FSI Server. You might want to specify a custom template if you want to retrieve IPTC data for all images.

MetaDataQueryParameters	
Description	HTTP Query to append to meta data requests
Syntax	Query
Default	—
Context	<Options>
Version	4.1.3 or higher

An optional query string of the form "a=1&b=2&c=3..." to append to meta data request to FSI Server. These query parameters will be appended to info and list requests passed to the imaging server.

ConcurrentRequests	
Description	Number of simultaneous image requests
Syntax	Number
Default	2
Context	<Options>
Version	2.3.0 or higher

Specifies the number of image details (image tiles) being requested simultaneously.

Please do not change this value unless you are certain that this is necessary and you are aware of possible consequences:

- Increasing this value might lead to increased imaging server load
- Decreasing the value will decrease the download performance

UseIntermediateZoomLevels	
Description	Use more zoom levels than contained in the source image
Syntax	Boolean
Default	true
Context	<Options>
Version	4.0.0 or higher

Images with resolution levels contain the same image in different resolutions. The size of the next level is twice the size of the preceding level. If you disable this option FSI Viewer accesses these resolution levels only. Otherwise image data will be retrieved more often.

UniqueID	
Description	Unique Identifier for plug-ins
Syntax	String (see description)
Default	—
Context	<Options>
Version	3.6.0 or higher

A unique string identifying the image / image collection displayed in FSI Viewer. You can for example use the path of the image or the name of a catalog. Do not use the following characters in UniqueIDs: [Space] ~ % & \ ; : " , < > ? #
The → *Notepad* plug-in for example requires a UniqueID value.

MouseVelocity	
Description	Continue mouse actions after mouse up
Syntax	Boolean
Default	false
Context	<Options>
Version	5.0.0 or higher

If you enable this parameter mouse actions like "pan" will be continued depending on the speed you moved the mouse. The higher the drag speed the longer the action will be continued. This option is especially useful for 3D views. With this parameter enabled and mouse mode set to "rotate 3D" the object will continue to spin rather than stopping immediately after mouse up. Additionally a very fast spin of a 3D view will add a blur effect to the object.

addConfigFileExtension	
Description	defines file extension to append to "cfg" parameter values
Syntax	String
Default	".xml"
Context	<Options>

This parameter defines the file extension to append to "cfg" parameter values. From FSI Viewer 6.x, file extensions will only be added automatically, if no 3 character file extension is present. E.g. when using "cfg=foo" the value will automatically be extended to "foo.xml" by default. When using "cfg=foo.txt" nothing will be added.

The extension to add can be defined by the "addConfigFileExtension" parameter. Set "addConfigFileExtension" to "false" to never add a file extension.

NOTE:

FSI Viewer versions before 6.0.0 ALWAYS add ".fsi" to "cfg" parameters, if the value does not already end with ".fsi".

8.10.10 Using Custom Language Files

In case you want to edit tool tips for existing languages or add support for additional languages you can edit or create language files in the subdirectory "/languages" of your FSI Viewer setup directory.

Each language file contains a list of phrases used in FSI Viewer user interface. The language files are XML files and need to be saved in UTF-8 format. The attribute "id" specifies the context of each phrase. Please note that you need to use XML entities for characters not allowed in XML files, like e.g.

&	for "&"
<	for "<"
>	for ">"
"	for "
'	for ,

You can use basic HTML-tags (see → *Required HTML-Source Code* to apply text formatting to the tool tips.

Excerpt from a language file

```
<Data language="english" color="000000" font="_ sans"
offsetTop="0" offsetLeft="0"
offsetWidth="0" offsetHeight="0">

    <tip id="ZoomOut"><b>Zoom Out</b><br/>Click to
decrease magnification</tip>
<tip id="ZoomIn"><b>Zoom In</b><br/>
Click to increase
magnification</tip>
<tip id="Downloading Data">Downloading:</tip>

[...]
```

You can duplicate existing language files or create new files as needed. To use custom language files, please use the "language" parameter and set the parameter value to the filename without the ".xml" extension.

8.11 Virtual 360° Presentations

Preparing 360° Presentations

In addition to zoomable 2D images FSI Viewer can be used to display virtual 360° presentations of an object. This way you can provide a photorealistic, zoomable and spinable view of an object. Creating 360° views requires multiple images of the same object from different angles.

There are two different methods to setup virtual 360° views:

- Using multiple source images
- Using a single, tiled source image



Note

You can have the imaging server setup the image list for you automatically based on a directory or search query. In this case you can choose "Publish to Web" => "Publish as FSI Viewer 360" from either a directory or search result.

Method A - Using Multiple Source Images

This is the recommended way to setup virtual 360° views.

Upload the images belonging to a 360° View to a subdirectory of your imaging server. Create a configuration file (e.g. "my3dview.xml") and add the list of source images belonging to this view like in the sample as shown:

Configuration for a 360° presentation as displayed in Figure A

```
<?xml version="1.0" encoding="$charset"?>
<fsi_parameter>

    <images3d prefix="my3Dimage/image_" suffix=".tif">
        <image path="01" />
        <image path ="02" />
        <image path ="03" />
        <image path ="04" />
        <image path ="05" />
        <image path ="06" />
    </images3d>
</fsi_parameter>
```

Please note that the sequence of the <image> nodes defines the sequence of the 360° scenes. You can use the parameter → *SceneSets* to modify the sequence by configuration later on.

You can either enter the complete path for each image or use the prefix and suffix attributes to have FSI Viewer assemble the file names from

prefix + imagepath + suffix

In the example above, the image path for the first image is therefore:
my3dimage/image_01.tif

Besides adding the <images3d> node to your configuration file you can alternatively load an external file containing the image list. This can be useful if you have a

server side script or database containing the image paths for 360° views. In this case

you only need to specify the images3dURL parameter which should contain the URL to the image list to retrieve.

Method B - Using a single, tiled source image

In this case you need to prepare the source image in the way, that it contains all required images as tiles of equal dimension. Please see the figure below which shows a so prepared source image with 12 image tiles.

The number of horizontal and vertical image tiles must be specified using the parameters → *TilesX* and *TilesY*.

Figure A displays a source image containing 12 tiles representing a rotation of 360° around the Y-axis in steps of 30°. In this case you simply add the parameters *TilesX*=3 and *TilesY*=4 to your FSI configuration.



Fig A: Rotation around Y-axis in 30° steps 4 x 3 = 12 tiles

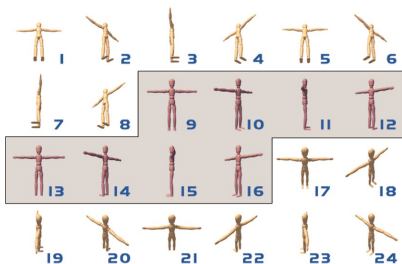


Fig B: Rotation around Y-axis in 45° steps and partial rotation around X-axis in 15° steps 6 x 4 = 24 tiles in 3 scene sets

Example – Configuration for a 360° presentation as displayed in Figure B

```
<?xml version="1.0" encoding="$charset"?>
<image>
  ...
  <TilesX value="4" />
  <TilesY value="3" />
  ...
</image>
```

Figure B shows a more complex 360° view representing a rotation of 360° around the X-axis in 45° steps and an additional rotation of +/- 15° around the Y-axis.

To present rotations around more than one axis or to present an object in different states you need to define → *SceneSets*.

A Scene Set describes a sequence of individual images.

The user can select individual images of a scene set using the keys "7" or "9" in the num pad or using the mouse mode "rotate" moving the mouse horizontally.

The user can additionally select a scene set using the keys "/" and "*" in the num pad or using the mouse mode rotate moving the mouse vertically.

If the TilesX or TilesY parameters are defined for a source image, but no scene set is defined, a scene set containing all the scenes from left to right, top to bottom will be created by default. Referring to Figure B this would be

```
<SceneSets value="1-24" />
```

As the presentation of Figure B consists of 6 by 4 individual images while 8 images represent a complete rotation around the Y-axis the definition of the following 3 scene sets is required:

Image Tiles	Y-Axis	X-Axis
1 – 8	0 to 360°	- 15°
9 – 16	0 to 360°	0°
17 – 24	0 to 360°	+15°

The required configuration to rotate the object around the Y-axis moving the mouse horizontally and rotating the object around the X-axis moving the mouse vertically looks like this:

Configuration for a 360° view as displayed in Figure B

```

<image>
  ...
  <TilesX value="6" />
  <TilesY value="4" />
  ...
</image>
<options>
  ...
  <SceneSets value="1-8;9-16;17-24" />
  ...
</options>

```

;

You can use the SceneSet parameter to specify the image sequence independent of the way you setup the 360° view (multiple source images or a tiled source image). Please refer to the description of the → *SceneSets* parameter for details on the different ways of defining images contained in a scene set.

Tip: If you are not sure about the number of images in your tiled source image in the x and y direction, try entering the value 1 for TilesX and TilesY. In the viewer you will then see the entire source image structure.

8.11.1 Parameters for Virtual 360° Presentations

The following FSI configuration parameters apply to virtual 360° presentations only.

Parameters for multiple source images (Method A) only:

Images3D	
Description	Node containing the list of source images
Syntax	XML node
Default	—
Context	ROOT node
Query	n/a

Add this node to the top level of a configuration file to list the individual images for a 360° presentation. You can add the attributes prefix and suffix instead of providing the complete path for each image individually.

Configuration for a 360° presentation as displayed in Figure A

```
<?xml version="1.0" encoding="$charset"?>
<fsi_parameter>

    <Images3d prefix="my3Dimage/image_" suffix=".tif">
        <image path="01" />
        <image path ="02" />
        <image path ="03" />
        <image path ="04" />
        <image path ="05" />
        <image path ="06" />
    </Images3d>
</fsi_parameter>
```

Images3DURL	
Description	URL to an XML file containing the list of source images
Syntax	URL
Default	—
Context	<options>

URL to an external XML file containing the image lists for 360° presentations as explained above. You might want to use the parameter Images3DURLBase in your _default.xml configuration file to specify the URL base for all 360° image list URLs.

Images3DPrefix, Images3DSuffix	
Description	Prefix / Suffix to add to the image URLs
Syntax	String
Default	—
Context	<options>
Version	4.0.7

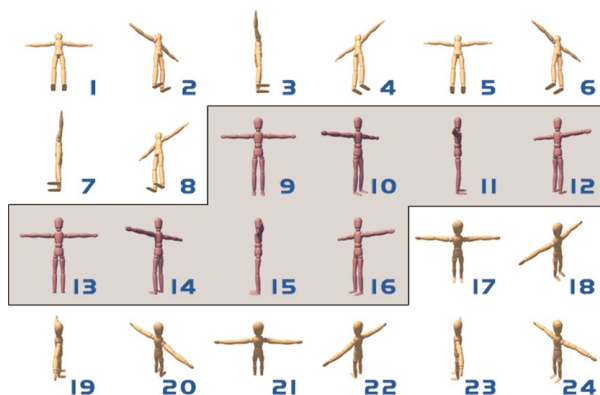
Prefix or suffix to add to the <image src=""> values. You might want to use these parameters to pass a prefix or suffix by HTTP query rather than using the corresponding "prefix" and "suffix" attributes in the <images3d> node.

Parameters for tiled source images (Method B) only:

TilesX and TilesY	
Description	Number of individual images for 360° presentations
Syntax	Number
Default	1
Context	<image> or deprecated <FPX>
Query	ImageTilesX and ImageTilesY

Number of individual images contained in the source image in x- and y- direction respectively.

Example: image containing 6 x 4 tiles arranged in 3 scene sets.



Even with 'asymmetrical' scene orders - like in the example to the left - the number of tiles would be 6 in the x direction and 4 in the y direction.

Definition of 3 scene sets containing 6 x 4 = 24 tiles

```

<image>
...
    <TilesX value="6" />
    <TilesY value="4" />
...
</image>
<options>
...
    <SceneSets value="1-8,9-16,17-24" />
...
</options>

```

Parameters for all 360° presentations (Method A or Method B):

SceneSets	
Description	Scene sequence definition
Syntax	String
Default	1 – n tiles
Context	<options>

Using this parameter you can define scene sets for 360° views. Scene sets define the order of scenes to be displayed.

The default is a single scene set containing all scenes (image tiles) from scene 1 to the last scene image. The default sequence is left to right, top to bottom in a tile image and the sequence of the <image> nodes for 360° views based on individual source images.

In the viewer scenes can be selected by moving the mouse horizontally and scene sets can be selected by moving the mouse vertically.

Syntax:

Individual sets have to be separated with ";" (semicolon).

Scenes within the sets are separated with "," (comma) and specify the 1-based index of the image from left to right, top to bottom up to the total number of scenes (TilesX x TilesY or individual source images).

Instead of individual scenes you can also specify scene ranges, e.g. "5-10".

If you leave out the start or the end of a scene range, 1 or respectively the last scene will be assumed in this case. If you specify a range using "->" the scenes

from the first number to the last scene image plus the first scene image up to the second number will be assumed. The example below illustrates the syntax:

Definition of multiple scenes and scenesets

```
<options>
...
    <SceneSets value="2,3,5-10;10-1;5-;8->3" />
...
</options>
```

The example defines 4 scene sets in total. Assumption: The source image is created from 10 individual images (scenes), 5 in a row, two rows. This leads to the following image order for the individual scenes:

Set 1 - Scenes: 2,3,5,6,7,8,9,10
 Set 2 - Scenes: 10,9,8,7,6,5,4,3,2,1
 Set 3 - Scenes: 5,6,7,8,9,10
 Set 4 - Scenes: 8,9,10,1,2,3

You can review the resulting scene sets in the "Info" section of the Debug Window.

ScenePreload	
Description	Load scenes in high resolution
Syntax	Boolean
Default	false
Context	<options>

If this parameter is activated (default) all scenes will be loaded at high resolution on startup. Otherwise preview images will be loaded with at a lower resolution. This parameter can significantly influence loading times and traffic volumes.

Setting this parameter to false leads to faster startup times and high resolution scene images will be loaded as soon as they need to be displayed.

TilePaddingX and TilePaddingY	
Description	Crops each scene image by a given number of pixel
Syntax	Number (pixel) or percent
Default	0
Context	<image> or deprecated <FPX>
Version	2.2.0 or higher

Specifies a positive number of pixels that each source scene image (2D and 360°) will be cropped by. This parameter can be used to eliminate image margins especially with 360° objects containing polychrome backgrounds. You can specify the padding either absolutely as a number of pixel or in percent of the source image.

NoSceneAnimation	
Description	No SceneSet animations
Syntax	Boolean
Default	false
Context	<options>

Disable animations between scenes and scene sets. This can be useful if different scenes represent different states of an object instead of a rotation.

NoSetLoop	
Description	No 360° rotation around the y-axis
Syntax	Boolean
Default	false
Context	<options>

The first and the last scene set will be considered ending points. (E.g. for partial rotations < 360°).
Please note: The blur animation is turned off if NoSetLoop is activated.

NoSceneLoop	
Description	No 360° rotation around the x-axis
Syntax	Boolean
Default	false
Context	<options>

The first and last scene in the each scene set will be considered ending points. (E.g. for partial rotations < 360°).

Please note: The blur animation is turned off if NoSceneLoop is activated.

8.12 Debug Mode

8.12.1 Enabling the Debug Mode

The debug mode can be enabled or disabled by passing the "Debug" parameter either by HTTP query or in an XML configuration file.

Please use "debug=1" to enable the debug mode and "debug=0" (or omit the debug parameter) to disable the debug mode. To disable the debug mode permanently you can safely delete the "/debug" directory of your FSI setup directory.

Example:

If FSI Viewer does not work or look as expected in an HTML page you can quickly debug FSI Viewer as follows:

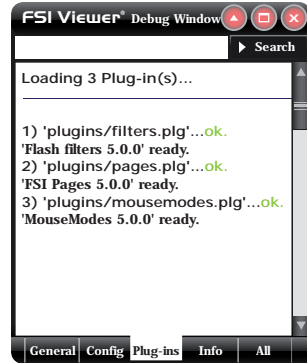
- View the source code of the HTML document and search for the FSI Viewer HTML Code
- Copy the entire FSI Viewer URL (<http://...fsi.swf?....>) to the clipboard
- Paste the URL to the address bar of your browser appending "&debug=1"

8.12.2 Using the Debug Window

After activating the debug mode an output window is available presenting information about the initialization progress, configuration parameters, plug-ins and image information.

You can select different debug sections by clicking the corresponding tab at the bottom of the window.

The window can be moved by dragging the window title bar and you can show or hide the window by pressing "d".



The section "All" contains the entire debug output. Please provide the content of this section when requesting support.

Enabling the debug mode additionally prevents the browser from caching FSI configuration files.

This allows you to easily test changes to your FSI configuration files without having to delete the browser cache each time. You can search each section by entering a keyword to the search input at the top. Pressing the "search" button multiple times will cycle through all matches.

General section

Provides general information about the initialization progress. If the viewer does not display an image you should be able to track down the error cause by determining the step where the initialization process stops.

For example if the last output is "Loading Skin and Module" the skin file is most likely missing or corrupt.

If the Viewer runs in evaluation mode you will be presented the reason why evaluating the licence key failed here. Please refer to the plug-in section for a list of active plug-ins and to the Info section for the domain evaluated by FSI Viewer.

Config section

Displays information on the configuration parsing process, the resulting configuration and possible configuration errors.

If the viewer loads successfully, but the configuration does not look as expected you should be able to track the reason in this section. This section lists all recognized parameters resulting from the 3 different configuration methods (default configuration, configuration file and query) and displays tips regarding false parameter values as well as unknown parameters.

Plug-ins section

Provides information on loaded FSI Plug-ins and configuration errors related to plug-ins. Additionally you can review all configuration parameters of the active plug-ins.

Info section

Displays version and licence information, general information related to the viewer instance and runtime information on the image being displayed.

Press "r" to refresh the image section being displayed after zooming, panning or rotating the image.

8.13 Automated Implementation of Images

Rather than creating individual configuration files for each image it is recommended to provide FSI parameters via HTTP queries appended to the FSI Viewer URL when implementing large numbers of images.

As some browsers require the obsolete <embed> tag, queries have to be added to both the <object> and the <embed> tag. (HTML-Source Code).

FSI Viewer is able to request image specific parameters like the source image dimension (parameters "Width" and "Height") from your imaging server (→ *Required HTML-Source Code*). Using this feature reduces the required work to dynamically providing the path to the source images.

It is recommended to use server side script like ASP, PHP, JSP, Perl, etc. when implementing large numbers of images. All subsequent code examples in this section refer to PHP, but can be easily adapted to any other server side scripting language.

Steps to be taken when using large numbers of images:

1. Defining default parameters

Be sure to provide as many recurring parameters as possible in the `_default.xml` file. This way you don't have to pass these parameters via query. The most important parameter is `ImageServer`, as this parameter enables you to subsequently define image parameters using relative addressing. If there are some images that require different parameter values than defined in the `_default.xml` file, you can still overwrite the default setting by passing the value via query.

2. Creating image specific URLs

Be sure to have image specific data available for server scripting by using a database, your content management system, or any other appropriate source. You will at least need to provide the `Image_Server` parameter from your data source. After collecting the image specific data from your data source you have to build a standard HTTP query string (RFC1738) containing the data using server side scripting (→ *Using HTTP Queries*).

Generating FSI queries (`fetch_fpx_image.php`)

```
// Retrieve FFX properties for picture 1 from table
"FPX_Images"
$result = mysql_query('select
FPXSrc,FPXHeight,FPXWidth,TilesX,TilesY
from FPX_Images where PictureID=1');
if ($result) {
// Fetch SQL data
$fpx_parameter=mysql_fetch_assoc($result);
if ($fpx_parameter){
// RFC1738 parameter encoding
foreach ($fpx_parameter as $fpx_key => $fpx_val) {
$query_collection[] = $fpx_key.'='.$fpx_val;
rawurlencode($fpx_val);
}
// Make one large query string
$fsgi_query=implode('&', $query_collection);
// Add Query to FSI Viewer URL
$fsgi_url='fsgi.swf?'.$fsgi_query;
}
}
```

3. Creating the HTML-Code dynamically

Create a variable that contains the `<object>` and `<embed>` tag with all variables provided as script variables. Replace all variables inside the

template using server side scripting. You will usually need to replace the following 4 variables:

- URL (see above: 2. Creating image specific URLs)
- Width
- Height
- Background Color

HTML Template

```
<HTML>
<BODY>
...

<?php
include("fetch_fpx_image.php");

$fsi_url =
"fsi/fsi.swf?FPXSrc=image.fpx&FPXTilesx=4&FPXTilesY=3";
$width = 320;
$height = 300;
$bgcolor = "#FFFFFF";

$template="
<Object      classid=\"clsid:D27CDB6E- AE6D- 11cf- 96B8-
444553540000\" codebase=\\
"http://download.macromedia.com/pub/shockwave/cabs/flash/s
wflash.cab#version=6,0,65,0\"
width=\"$width\" height=\"$height\">
  <PARAM name=\"movie\" VALUE=\"$fsi_url\" />
  <PARAM name=\"bgcolor\" VALUE=\"$bgcolor\" />
  <PARAM name=\"menu\" VALUE=\"false\" />
  <PARAM name=\"allowscriptaccess\"
value=\"always\" />
  <PARAM name=\"allowfullscreen\" value=\"true\" />
  <PARAM name=\"quality\" value=\"high\" />
<EMBED TYPE=\"application/x-shockwave-flash\"
PLUGINSPPAGE=\"http://www.macromedia.com/shockwave/download
/index.cgi?Pl_
Prod_ Version=ShockwaveFlash\" SRC=\"$fsi_ url\"
WIDTH=\"$width\" HEIGHT=\"$height\"
BGCOLOR=\"$bgcolor\" MENU=\"false\"
ALLOWSCRIPTACCESS=\"always\" ALLOWFULLSCREEN=\"true\"
QUALITY=\"high\"></EMBED>
</Object>";
```


HTML Template

```
echo $template;
?>

...
</BODY>
</HTML>
```

8.14 Plug-ins

FSI Plug-ins dynamically extend the functionality of FSI Viewer and can be included at runtime.

Depending on the plug-in additional buttons might be added to the menu bar. The sequence of these buttons results from the sequence of the plug-in definitions inside the <Plugins> section of the configuration file.

Using FSI Viewer Plug-ins

To include a plug-in you simply list the desired plug-ins in the <Plugins> section of any *.xml configuration file (or the _default.xml file).

The required plug-in files are located in the "/plugins" folder of the FSI setup directory.

Using FSI Plug-ins

```
<fsi_parameter>
...
<Plugins>
  <Plugin src="mousemodes" />
  <Plugin src="history" length="10" />
</Plugins>
...
</fsi_parameter>
```

The "Src" attribute which applies to all plug-ins specifies the plug-in to be included:

Src	
Description	Name of the plug-in
Syntax	String
Default	—
Context	<Plugins>
Version	2.0.0 or higher

This attribute defines the name of the plug-in, e.g. "history". The value has to be stated without the file extension ".plg".

Alternatively you can integrate plug-ins by HTTP query using the "plugins" parameter with a comma separated list of the desired plug-ins.

Plug-in Parameter

```
http://www.example.com/fsi.swf?...&plugins=mousemodes
```

This option is available from FSI version 3.0 only.

Param

```
<PARAM  
name="movie" VALUE="/fsi/fsi.swf?plugins=mousemodes,history">
```

The example above integrates the plug-in → *Mousemodes* and the plug-in → *History*

8.14.1 Deactivating Plug-ins

To disable previously integrated plug-ins (e.g. plug-ins defined in the _default.xml file) you can set a parameter with the name of the plug-in and a value of "false" to the <Options> section of your configuration file or provide a corresponding parameter via HTTP query.

Disabling the History Plug-in

```
<Options>  
...
```

Disabling the History Plug-in

```
<History value="false" />
...
</Options>
```

or by HTTP query:

Using a Query to disable History Plug-in

```
http://www.example.com/fsi.swf?...&history=false
```

8.14.2 Defining Plug-in Parameters

Plug-in parameters are usually defined as attributes of the `<Plugin>` tag.

Please note that some plug-ins might require specific attributes or even entire XML sections in the configuration file to work properly. For information on a specific plug-in please refer to the → *Plug-in Reference*.

Defining plug-in parameters

```
<Plugins>
...
  <Plugin src="navextension" MenuOffset="10"
    visible="true" />
  <Plugin src="zoommeter">
    <SRCRelative value="true"/>
  </Plugin>
...
</Plugins>
```

8.14.3 Defining Plug-in Parameters dynamically

To provide plug-in parameters by query you have to prefix the corresponding parameter with the name of the plug-in and an underscore character:

PluginName_ParameterName=value

Providing plug-in parameters via HTTP query

```
fsi.swf?navextension_visible=true&history_length=10
```

The example above sets the "visible" parameter of the "NavExtension" plug-in to "true" and the "length" parameter of the "History" plug-in to "10".

Similar to providing plug-in parameters you can provide plug-in parameters in the options section of FSI Viewer configuration files by prefixing the parameter name with the name of the plug-in. This way you can specify plug-in parameters without actually including the plug-in again.

Providing plug-in parameters as global parameters

```
<Options>  
...  
    <history_length value="10" />  
...  
</Options>
```


9 FSI Pages Add-on

Using FSI Pages you can present image collections online simulating a printed catalog, booklet or photo album. FSI Pages automatically creates a page index with thumbnails and provides interactive zooming using FSI Viewer.



The number of pages (images) has a very low impact on the loading time and performance. You can therefore use FSI Pages with 1,000 and more images or pages. Depending on the configuration, FSI Pages can be used to display different types of booklets like catalogs (with or without links), booklets, photo albums and more.

9.1 Using FSI Pages

The integration of FSI Pages is very similar to an FSI plug-in. To enable FSI Pages you have to add a plug-in node to your XML configuration file:

Integrating FSI Pages

```
<Plugins>
...
    <Plugin src="pages" skin="silver" />
...
</Plugins>
```

Parameters for FSI Pages can be provided by adding parameters as attributes or child nodes of the FSI Pages plug-in node. As with all FSI plug-ins, you can additionally provide parameters by adding the parameter to the FSI Viewer query prefixed by "pages_".

Providing parameters as node attributes

```

<Plugins>
...
    <Plugin src="pages" MarginLeft="100" zoom="false" />
...
</Plugins>

```

Providing parameters as child nodes

```

...
<Plugins>
...
    <Plugin src="pages" MarginLeft="100" zoom="false" >
        <MarginLeft value="100" />
        <zoom value="false" />
    </Plugin>
...
</Plugins>
...

```

Providing parameters using HTTP queries attached to the movie URL

```
fsi.swf?cfg=foo&pages_marginleft=100&pages_zoom=true
```

9.2 FSI Pages Converter

FSI Pages Converter is a Java based conversion tool to split PDF documents into separate TIFF files optimized for use with FSI Server.

Besides converting PDF documents into individual TIFF files, the conversion tool automatically extracts hyperlinks from the PDF document and generates partial PDF documents of the document for download. Please contact your FSI dealer to get a copy of FSI Pages Converter.

9.3 Aspect Ratio of the Pages

By default FSI Pages uses the aspect ratio of the first image in the image collection to determine the aspect ratio for all pages.

Alternatively you can manually define an aspect ratio for all pages using the "ratio" parameter.

E.g. `<Plugin src="pages" ratio="1:2" />`
 or `<Plugin src="pages" ratio="640:480" />`

9.4 Layout and Skins

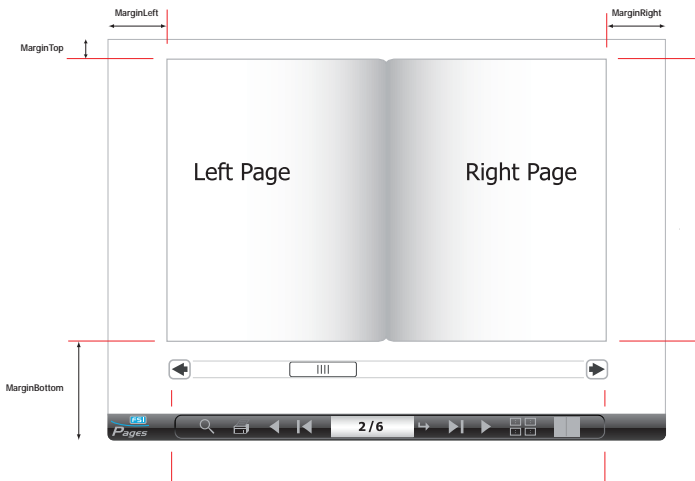
FSI Pages ships with two default skins that correspond to the default skins of FSI Viewer and an additional skin providing the user interface in a side bar. You can easily change the FSI Pages skin being used by specifying the file name of the skin using the FSI Pages "skin" parameter e.g.

Setting the skin

```
<Plugin src="pages" skin="silver" />
```

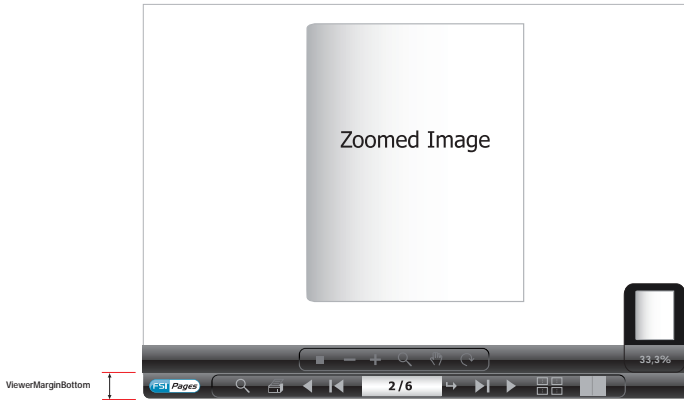
Additional custom skins are available on demand and may contain additional functionality and designs.

You can modify the layout of FSI Pages and FSI Viewer using the `MarginBottom`, `MarginLeft`, `MarginRight`, `MarginTop` and `ViewerMarginBottom`, `ViewerMarginLeft`, `ViewerMarginRight`, `ViewerMarginTop` parameters. The following illustrations show the use of these parameters:

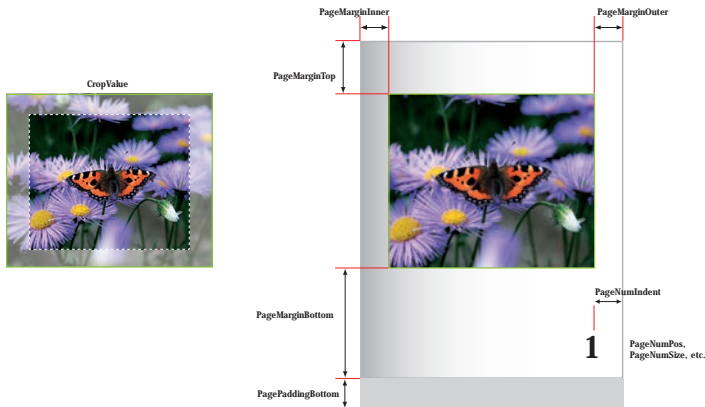


Using Margin parameters to define the position of the pages and to provide space

for the FSI Pages user interface.



Using *ViewerMargin* parameters to define the size and position of FSI Viewer inside FSI Pages. Usually you only specify *ViewerMarginBottom* to provide space for the FSI Pages interface.



9.5 Cropping

CropRect

The CropRect parameter defines a cropping rectangle applied to the source image. The four points to define this rectangle are floating point numbers ranging from 0.0 to 1.0, representing the edges of the image. The rectangle is defined by four floats, the order is Left, Top, Right, Bottom. The default rectangle is "0,0,1,1" which displays the entire image without any cropping. The rectangle "0,0,0.5,0.5" will crop the image to the top left quarter while "0.5,0.5,1,1" will crop to the bottom right quarter.

If set, the CropRect aspect ratio will define the pages aspect ratio in FSI Pages. If CropValues (see below) are defined, the CropRect will be ignored.

CropValues

In contrast to CropRect above, the CropValues can also be given in pixel or percent. CropValues are used to define a certain area of the source image to be cut off. E.g. the CropValue "10px,20px,5%,10%" will remove 10 Pixel from the left and 20 Pixel from the top of the image while also removing 5% from the right and 10% from the bottom. If CropValues are defined, any given CropRect will be ignored.

Margin/Padding

Page margins can additionally be defined to adjust the appearance of the images in FSI Pages. A page margin will hide the part of the image as defined, while padding will add some white space to the pages. Please refer to the image (XYZXYZYZ)

AutoCropPages

This will crop each image to fit the pages aspect ratio automatically.

ImageAlign

Mostly used for photo albums with FSI Pages. If the source image does not fill the entire page, this defines the alignment of the image. Default value is "TC" which stands for Top Center. Possible values are all combinations of the vertical alignments (Top, Center, Bottom) and horizontal alignments (Left, Center, Right).

A complete reference of all → *Parameters defining Layout and Appearance* can be found at the end of this chapter.

9.6 Defining Image Collections

The images to be displayed in FSI Pages have to be defined in an <images> node added to your XML configuration file. The format and the options of the collections are the same as those of the → *11 FSI Showcase Add-on* collections.

There are three different ways of defining image lists:

1. Retrieving automatically generated image lists from FSI Server The most convenient way to create image collections is using the template system of

FSI Server. You can either do so by using the "Publish to Web" option in FSI Server web interface or by specifying the server address and a directory or search query manually.

Retrieving Image Collections from Imaging Server

```
<PLUGIN src="pages">
  <ListTemplate value="catalog_list" />
  <dir value="images/catalogs/foo" />
</PLUGIN>
```

This way FSI Server uses an image → *ListTemplate* to return the matching images in XML format. Depending on the list template, FSI Server serves IPTC data belonging to an image (e.g. for links) as well.

2. Retrieving Directory based Image Collections from FSI Server

Retrieving Directory based Image Collections from FSI Server

```
<Plugins>
...
  <!-- Passing a directory as plain text -->
  <Plugin src="pages" server="..."
  dir="images/flowers/" />
  <!-- Passing an encrypted string representing a
  directory -->
  <Plugin src="pages" server="..."
  dir="*XzPzLUFVSHtmeeFxOY0NUU0" />
...
</Plugins>
```

3. Creating collections manually using XML configuration files

Creating collections manually using XML configuration files

```
<Images>
...
  <Image file="image1.xml"
  label="My First Image" />
  <Image file="image2.xml"
  label="My Second Image" />
...
</Images>
```

Method 2 and Method 3 can be freely combined while Method 1 retrieves a complete image list ignoring previously defined <images> nodes.

Image Order

The order of the images (this is the page order) corresponds to the order of the <image> declarations within the <images> section.

Inserting Blank Pages

You can insert blank pages by adding <image empty="true" /> nodes anywhere in your image collection. Alternatively you can use the FSI Pages parameter → *EmptyImages* to pass the page numbers to insert blank pages before separated by commas. This is especially useful if you retrieve the image collection from FSI Server (Method 1).

Removing certain Images from a List

Using the parameter "RemovePages" you can disable (remove) certain images from an image list. In this case you pass the page number(s) of the pages you want to remove separated by commas. This is especially useful if you retrieve the image collection from FSI Server (Method 1).

9.7 Page Reading Order

By default FSI Pages displays pages in western reading order: left-to-right. For documents with right-to-left reading order (e.g. Arab documents) you can change the reading order to RTL (right-to-left) using the boolean FSI Pages parameter → *PageOrderRTL*.

Defining Right-To-Left reading order

```
<Plugins>
...
    <Plugin src="pages" >
        <PageOrderRTL value="true" />
    </Plugin>
...
</Plugins>
```

Changing the page order to RTL changes all aspects of FSI Pages related to the page sequence and page numbering including the page sequence, the page numbering and the thumbnail display.

Please note that the page addressing does not change so that e.g. setting the → *InitialPage* parameter to "10" shows the contents on startup independent of the →

PageOrderRTL parameter. The same applies to "#page=n" hyperlinks in PDF documents.

If you just want to reverse the page sequence without changing the reading order, you can use the boolean FSI Pages parameter → *ReversePageOrder*

9.8 Printing Pages

If you want to enable users to print pages from within FSI Pages you can enable the built-in print function by setting the FSI Pages parameter → *Print* to "true".

Enabling Print Options

```
<Plugins>
...
    <Plugin src="pages" >
        <Print value="true" />
        <PrintResolution value="1024" />
    </Plugin>
...
</Plugins>
```

This adds a "Print" button to the menu bar. Clicking the "Print" button displays the "Print" dialog of FSI Pages. The → *Print* dialog offers different printing options to the user, depending on the current page and if the user zoomed into a page.

The printing options cover:

- Selected Range (e.g. 1-3; 9-20 or 1,6,8,12)
- Both Pages on a single sheet of paper
- Left Page
- Right Page
- The currently zoomed image section

After the user has chosen an option and clicks the "Print" button in the dialog, FSI Pages downloads the required image data (pixel data) from the imaging server and opens the print dialog on the user's system once the download has finished.

You can limit the resolution used for printing using the FSI Pages parameter → *PrintResolution*. Lower resolution leads to faster downloads at the cost of decreased print quality.

You can additionally pass effect parameters → *PrintEffects* to sharpen the image or to specify the image compression. You might additionally want to use a specific real time template used to retrieve the image for printing. Please use the FSI Pages parameter → *PrintTemplate* in this case. You can limit the displayed options of the "Print" dialog using the FSI Pages parameter → *PrintOptions*.

9.9 Saving Pages

If you want to enable users to download a file related to the displayed page, the current section or the entire document, you can enable the save options by setting the FSI Pages parameter → *Save* to "true".

Enabling Save Options

```
<Plugins>
...
  Plugin src="pages" >
    <Save value="true" />
    <SaveAllowSavingImages value="true" />
  </Plugin>
...
</Plugins>
```

Without any additional preparation this allows users to download an image of the left or right page.

9.10 Saving PDF or other document types

In addition to downloading images you might want to enable users to download the complete source PDF or specific parts of the source PDF documents. This documentation refers to PDF documents only, but you can also provide other file types for download.

Downloading the entire PDF Document

Enabling this option is as easy as adding the FSI Pages parameter → *Advanced Parameters* providing the absolute URL to a PDF document to download.

Example:

```
<SaveDocumentFile value="http://foo.com/somepdf.pdf" />
```

Downloading single or double pages

In order to enable the user to download a PDF document containing the page currently viewed in FSI Pages, the XML based data for a page needs to contain the corresponding URL or file name to that document.

If you are using FSI Pages Converter to convert your PDF document, you only need to enable the "PDF Output" option in FSI Pages Converter and FSI Pages Converter splits your source document into multiple, partial PDF documents. You will then need to upload these files to some web server to provide them for download and add the FSI Pages parameter → *SaveURLPrefix* and *SaveUrlSuffix* specifying the URL to the directory on your server, containing these files.

Example:

You uploaded the PDF documents created by FSI Pages converter to your server and the URL to download the files is e.g.

http://www.foo.com/downloads/catalog1/*.pdf

For this example your configuration should look like this:

```
Enabling Save Options
<Plugins>
...
  <Plugin src="pages" >
    <Save value="true" />
    <SaveDocumentFile value="
      http://www.foo.com/downloads/
      catalog1/all.pdf" />
    <SaveURLPrefix value="
      http://www.foo.com/downloads/catalog1/" />
  </Plugin>
...
</Plugins>
```

Downloading parts of the Document

If your document is separated into logical parts (e.g. chapters or product groups), you might want to enable the user to download parts of the document. In this case you need to use the FSI Viewer plug-in → *Chapters* which enables the user to select and go to a predefined section in your document. For each section you can specify a separate document to offer for download. This way the user will be presented an option to download the current section (chapter) in the "Save" dialog.

```
<Indexdata>
...
  <Index page="1" label="Chapter 1"
    file="chapter1.pdf" />
  <Index page="20" label="Chapter 2"
    file="chapter2.pdf" />
  <Index page="30" label="Chapter 3"
    file="chapter3.pdf" />
...
</Indexdata>
```

Please note:

If you use the FSI Pages parameters → *SaveURLPrefix* and *SaveUrlSuffix* all "file" attribute values will be prefixed and/or suffixed with the values you specified. This way you can enter the file names only and provide the path to the files using the FSI Pages parameter → *SaveURLPrefix* and *SaveUrlSuffix*.

9.11 Searching in FSI Pages

In FSI Pages provides an optional search dialog to perform full text search in documents. For performance reasons FSI Pages does not perform the actual search itself. Instead it accesses FSI Server or any other server capable of returning search results in XML format.

To enable the search dialog in FSI Pages you simply need to set the FSI Pages parameter "Search" to true. This will add a button to FSI Pages to show or hide the search dialog. Depending on the dimension of the FSI Pages instance you might want to increase the size of the search dialog using the FSI Pages parameters → *SearchDialogHeight* and *SearchDialogWidth*.

By default FSI Pages adds wildcards "*" to the search keywords the user entered. You can disable this feature in case you prefer displaying exact matches or if you are using an external server to return search results that performs wildcard searching by default.

Using FSI Server to return Search Results

By default FSI Pages uses FSI Server to retrieve search results. FSI Pages passes the term entered by the user to FSI Server and receives an XML based list of matches. Based on these matches FSI Pages presents a list of thumbnails of those pages matching the search criteria.

How does FSI Server determine matches?

By default FSI Server searches the IPTC data sets that FSI Pages Converter stores the keywords in only: `iptc.caption` and `iptc.fsi_search_data`.

If your images do not yet contain keywords you might want to add keywords to the corresponding IPTC data sets of your images. If you are using a PDF document as a source for your FSI Pages instance you can have FSI Pages Converter version 2 create IPTC keyword data for all pages based on the content of the PDF document. Please refer to the "Text Search" section of FSI Pages Converter in this case.

In case you want FSI Pages to search all (or other) IPTC data sets, please use the FSI Pages parameter "SearchQueryTemplate". The default value when using FSI Server is:

```
iptc.caption:%%searchtext%%%20iptc.fsi_search_data:%%searchtext%%
```

which limits the search to the two given IPTC sets. To search in all data sets available, modify the parameter value to: `%%searchtext%%`

Using external Servers to return Search Results

Alternatively to using imaging server you might want to use an existing database on an external server to perform the search. In this case FSI Pages provides a simple interface you can use to have FSI Pages direct the search to and retrieve the results from the external server.

This is by nature an advanced feature and requires some server side scripting and database knowledge. The following section describes the required steps based on the assumption that your database contains a unique ID for each FSI Pages instance (catalog) and the required keywords.

Directing the search query to your server side script

Please use the FSI Pages parameter `→ SearchCustomURL` to provide the URL to the script on your server performing the search and returning the XML based results. To identify the FSI Pages instance (catalog) to return results for, you might need to pass a value identifying the catalog. This can be done using the FSI Pages Parameter `→ SearchCustomValue`. Both parameters can – as usual – be specified by query or by adding the parameter to a configuration file.

Configuration parameters for external searching

```
<Plugins>
...
  <Plugin src="pages" >
    <SearchCustomURL value=
      "http://foo.com/search.php" />
    <SearchCustomValue value="12345" />
  </Plugin>
...
</Plugins>
```

Using the parameters like in the example above will force FSI Pages to direct the search to the URL specified using the HTTP POST method. The POST data contains the following values:

searchvalue: The search text entered by the user
customvalue: The value you specified using "SearchCustomValue"

You might want to pass multiple, comma separated values to "SearchCustomValue" in case you need multiple values to identify a given FSI Pages instance.

Returning the XML based list of matches to FSI Pages

Your script needs to return the search result in XML format. Each result must be returned as a child node of the root node <result> (see example below). Each result item must at least contain the page number corresponding to the match. Additionally each result item may contain HTML formatted text describing the match. The sequence of the result item nodes specifies the sequence of the results displayed in FSI Pages search dialog. You therefore might want to sort the items by relevance before writing the matches to the list. Please use utf-8 encoding in case the result contains language specific, special characters.

Search Result (XML)

```
<?xml version="1.0" encoding="utf-8"?>
<Results>
  <Item page="10">
    Some <b>text</b> for this result...
  </Item>
  <Item page="20">
    Some <b>text</b> for this result...
  </Item>
  <Item page="30">
    Some <b>text</b> for this result...
  </Item>
</Results>
```

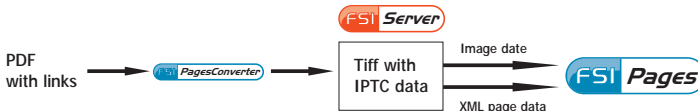
9.12 Hyperlinks

You can add interactive areas (links) on pages by providing XML data for each page specifying the coordinates, the tool tip and the action to execute.

You might for example want to publish a PDF document containing links. In this case you can use FSI Pages Converter to extract the links from the PDF document and create the required XML data for FSI Pages for you.

9.13 Publishing PDF Documents with Links using FSI Pages Converter and FSI Server

The easiest way to add links to FSI Pages is using a source PDF document containing hyperlinks. This way FSI Pages Converter prepares the required XML data by gathering the link information from the PDF document. FSI Server subsequently serves the XML data to FSI Pages.



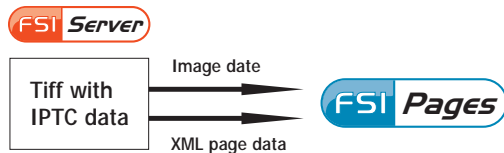
1. Convert the source PDF document using FSI Pages Converter. Make sure to check the «Include Links» option in the "Hyperlinks" option section. This way FSI Pages Converter extracts link information from the PDF document. Make sure to uncheck the "Enable XML Output" option in the "XML Output" option section. This way FSI Pages Converter writes the XML data directly to the IPTC data of the converted TIFF files.
2. Upload the TIFF files (containing the IPTC data) to a directory on FSI Server
3. Go to FSI Server Interface and select the directory containing the TIFF files
4. Select "Publish to Web", "Publish as FSI Pages" Make sure to use a configuration preset with links, e.g. "Catalog with links".

9.14 Providing XML page data to FSI Pages

There are three alternative ways to provide XML data like links for each page.

a) Using FSI Server and IPTC data fields

FSI Server is capable of serving the IPTC data contained in an image file to FSI Viewer (FSI Pages). By default FSI Pages Converter adds the



required IPTC data to the output TIFF files. The IPTC data for each page can then automatically be retrieved by FSI Pages using the FSI Server template system. This

is the way it works when using "Publish as FSI Pages with links" in FSI Server web interface. A great advantage of this option is, that the data persists even when renaming or moving images.

For imaging servers not capable of serving IPTC data to FSI Pages, you need to choose one of the alternative methods below.

b) Using individual XML files

An alternative way to provide XML page data is using external XML files. With this method each image requires a related XML file containing the page data stored in a directory on your web server. The XML files have to be named after the related images with the additional suffix ".xml".



Example:

The XML file name for the image "page065.tif" must be "page065.tif.xml".

Usually FSI Pages Converter creates the required files for you when converting a PDF document. Please enable the option "XML Output" of FSI Pages Converter in this case.

Example: You converted a PDF document "catalog1.pdf" using FSI Pages Converter and enabled the "XML Output" option. In the output directories you specified in FSI Pages converter you can now find the TIFF files and the related XML files, e.g.

pagee0001.tif	page0001.tif.xml
page0002.tif	page0002.tif.xml

Now you need to upload the images to your imaging server and upload the XML files to a directory on your web server.

Finally you need to provide FSI Pages with the information where the XML files can be found on your web server. Please add the Parameter → *XMLBase* to your configuration for this purpose.

Say the XML files can be accessed via:

```
http://www.myserver.com/xmldata/page01.tif.xml
http://www.myserver.com/xmldata/page02.tif.xml
...
```

The corresponding parameter for FSI pages looks as follows:

```

XML Page data provided by XML files
<Plugins>
...
  <Plugin src="pages" >
    <XMLBase value="
      http://www.myserver.com/xmldata/" />
  </Plugin>
...
</Plugins>

```

This way FSI Pages reads an XML file for each page from the location specified.



Please note that the XML files have to reside in the same domain as FSI Viewer.

c) Using <pages> nodes in configuration files

The third method is adding a <pages> node to the configuration file used for a page. Note that this method requires an individual configuration file for each page.



The example below shows the structure of page data added to a configuration file:

XML Page data in configuration files

```
<fsi_parameter>

<FPX>
...
</FPX>

<Pages>
  <Page file="page065.tif.xml">
    <Links>...</Links>
  </Page>
</Pages>

</fsi_parameter>
```

XML Data Format for Pages

Usually you don't need to edit the XML data for FSI Pages manually, as FSI Pages Converter creates the XML data for you. The description below provides information on the format in case you want to edit the data manually or if you want to use a different tool to create the page data.

XML Page data with 2 links

```
<Page file="page16.pdf">
  <Links urlprefix="http://somesite.com/product.php?id=">
    <Area tip="Some Tip" url="599">
      <Shape coords="0.20,0.73, 0.25,0.73,
0.25,0.72, 0.2,0.72" />
      <Shape coords="0.35,0.73, 0.41,0.73,
0.41,0.72, 0.35,0.72" />
      <Text>Some long descriptive text</Text>
    </Area>
    <Area tip="Another Tip" url="210">
      <Shape coords="0.20,0.73, 0.25,0.73,
0.25,0.72,0.2,0.72" />
    </Area>
  </Links>
</Page>
```

<page> node

This is the root node for XML page data. The following attributes will be recognized by FSI Pages:

Attribute	Description
File	URL or file name to use for the → <i>Saving Pages</i> functionality of FSI Pages"

<page> node

```
<page file="page16.pdf" />
```

<links> nodes

These nodes encapsulate one or more <area> nodes. Each <area> node represents a link. You can use the following attributes to specify default values for all child <area> nodes. Using multiple <links> nodes you can use different default values for the <area> nodes contained.

Attribute	Description
URLPrefix*	Prefix to add to all URLs specified in the child <area> nodes
URLSuffix*	Suffix to add to all URLs specified in the child <area> nodes
DefaultURL	Default URL to use for <area> nodes without a "URL" attribute
DefaultTarget	Default target frame for <area> nodes without a "target" attribute
DefaultTip	Default tool tip for <area> nodes without a "tip" attribute

* The prefix and suffix will be used for relative links only.

Special links beginning with "javascript:", "mailto:" or "#page=" will be skipped as well.

<links> node

```
<Page>
  <Links URLPrefix="http://foo.com/" DefaultTip="No Tip">
    <area>...</area>
  </Links>
</Page>
```

<area> nodes

This node represents a link on a page.

Each link must contain at least one <shape> node describing the link area. The table below lists the attributes recognized by FSI Pages.

Attribute	Description
URL	URL to open when the user clicks the link.
Target	Target frame to open the URL in
Tip	Tool tip to display when the user points at the link

<area> node

```
<Page>
  <Links>
    <Area URL="info.html" URLTarget="_blank" Tip="Hello" >
      ...
    </Area>
  </Links>
</Page>
```

<shape> nodes

These child nodes of the <area> node provide the coordinates of a link polygon. The format for the coordinates is "x1,y1, x2,y2, ...". The coordinates are floating point values between 0 and 1, where 0 is the left side of the image and 1 is the right side of the image. For y-coordinates 0 is top of the image and 1 is the bottom of the image. A valid shape must at least consist of 6 values (3 points).

You can use multiple <shape> nodes in an <area> node if you want to specify multiple shapes for the same link.

Attribute	Description
Coords	Comma separated coordinates describing the link shape (see above)

<shape> node

```

<Page>
  <Links>
    <Area>
      <Shape coords="0,0, 1,0, 1,1, 0,1,
        0,0" />
    </Area>
  </Links>
</Page>

```

The example describes a rectangular shape filling the entire image.



Please provide larger shapes first when providing overlapping shapes. FSI Pages stacks shapes above each other in the order they have been defined.

You can use any closed polygon shape. Please note that complex polygons might lead to reduced page flip performance. FSI Pages will automatically close non-closed polygons by connecting the last point to the start point.

<text> nodes (optional)

Each <area> may contain a <text> node providing HTML formatted text providing a description for the link. The → *Large ToolTips* plug-in uses the content of this node to display tool tips for hyperlinks.

<shape> node

```

<Page>
  <Links>
    <Area>
      <Shape coords="0,0, 1,0, 1,1, 0,1" />
      <Text><b>Some descriptive text</b>
    </Text>
    </Area>
  </Links>
</Page>

```

9.15 Link Parameters and Hierarchy of Link Parameters

FSI Pages provides a number of ways to define link properties. This provides a flexible way to configure links without the need to change the page XML data, but

adds a certain level of complexity.

This chapter describes how FSI Pages assembles link URLs and the hierarchy of link parameters in general.

Example: Hierarchy of "LinkURL" parameters and attributes

Parameter	XML Attribute	Location	Description
ForceLinkUrl		FSI Pages parameter	Force this URL for ALL links
	URL	<area> node	URL for this link
	DefaultURL	<links> node	Default URL for all child links
DefaultLinkUrl		FSI Pages parameter	Default URL for links

FSI Pages checks if there is a global FSI Pages Parameter "ForceLinkURL" first. In this case all link URLs will be overwritten with this parameter.

If the **<area>** node does not contain an URL attribute, FSI Pages uses the DefaultURL attribute of the parent **<links>** node. If this attribute is undefined, FSI Pages uses the global FSI Pages Parameter "DefaultLinkURL".

The same process applies to the "target", "tip", "prefix", "suffix" and "javascripttarget" parameters:

Force-Parameter	Default-Parameter	<links> node	<area> node
ForceLinkURL	DefaultLinkURL	DefaultURL	URL
ForceLinkTarget	DefaultLinkTarget	DefaultTarget	Target
ForceJavascriptTarget	DefaultJavascriptTarget		Target
ForceLinkTip	DefaultLinkTip	DefaultTip	Tip or <text> child node
ForceLinkURLPrefix	DefaultLinkURLPrefix	URLPrefix	n/a
ForceLinkURLSuffix	DefaultLinkURLSuffix	URLSuffix	n/a

9.16 Relative and Absolute Links

FSI Pages does not add prefixes or suffixes to absolute link URLs. In this case any prefixes and suffixes will be ignored.

Absolute Link URL
<pre><Page> <Links> <Area URL="http://www.fsi-viewer.com">... </Area> </Links> </Page></pre>

Resulting URL: <http://www.fsi-viewer.com>

The same applies to → *Special URL Values* like "javascript:" and "mailto:" as well as "FSI Pages command URLs" beginning with "#". For relative links FSI Pages adds the URLPrefix and/or URLSuffix values if specified.

Relative Link URL
<pre><Page> <Links URLPrefix="http://foo.com/" URLSuffix="&ttest=1"> <Area URL="products.html?id=123">...</Area></pre>

Relative Link URL
</Links> </Page>

Resulting URL: <http://www.foo.com/products.html?id=123&test=1>

9.17 Special URL Values

For some special URL values FSI Pages ignores the URLPrefixes and URLSuffixes as well. The following table lists these special URLs:

URL	Description
javascript:foo()	Call the JavaScript function foo()
mailto:email@domain.tld	Send an email to email@domain.tld
#none	Do nothing. Handy if you just want to display a tool tip. Available in FSI Pages version 4.0.7 or above
#page=n	Forces FSI Pages to flip to page n
#showimage=[imageURL]	Display a custom JPEG image. The value [URL] needs to be an absolute URL to a JPEG image.
#zoom	Forces FSI Pages to zoom to the area covered by the link's shape(s)
#zoomarea	Magnifies the area covered by the link's shape(s) in place
#zoom=1,1,0,0,0.5,0.5	Forces FSI Pages to zoom to the image area following "=" (See Parameter → <i>InitialView</i> for format details)
#zoomonpage=n;strView	Combination of "#page=" and "#zoom=". This is: first flip to page n, then zoom to area strView. Available in FSI Pages version 4.0.7 or above

Please note that the special links starting with "#" need to be provided exactly as listed in the table above. Using e.g. "#zoomOnPage=[...]" or "foo.html#zoom" will link to the HTML anchor provided rather than executing an FSI Pages action.

For "mailto:" URLs the target frame is always "_self", except the XML <area> node specifies a different target.

9.18 Modifying Link Values at Runtime

In some cases it might be necessary to replace parts of link URL values for links on the pages – for example when using session variables or if you use the same catalog in HTTP and HTTPS environments. Runtime link template replacement applies to links on pages as well as for "SaveURL" values used to specify external document related to a page.

You can replace any part of these URLs using the FSI Pages parameters → *LinkTemplates* and → *LinkTemplateData*.

Use the FSI Pages parameter "LinkTemplateData" to specify a comma separated list of values you want FSI Pages to replace the keywords with. Usually you will provide the LinkTemplateData value by appending the parameter to the FSI Viewer query using server side script.

When specifying multiple LinkTemplates, the number and sequence of the corresponding LinkTemplateData values must match to achieve correct replacement.

For example replacing the templates "[SESSION]" and "[DATE]" at runtime in all link URLs can be done like this:

Replacing parts of link URLs at runtime

```
Configuration:
...
<Plugins>
...
    <Plugin src="pages" >
        <LinkTemplates value="[SESSION],[DATE]" />
    </Plugin>
...
</Plugins>
...
FSI URL
.../fsi.swf?LinkTemplateData=somesession,01.01.2010
```

This will replace all instances of "[SESSION]" by "somesession" and all instances of "[DATE]" by "01.01.2010" in all link URLs.

Please note that LinkTemplateData values have to be provided url-encoded (see → *Escaping or URL-encoding parameter values* if the values contain invalid HTML query characters like "?&/,").

Besides using the LinkTemplates parameter to provide custom templates, you can use the following predefined LinkTemplates without providing the corresponding LinkTemplateData:

Template	Description
_FSI_CURRENTPAGE_	1-based index of the current page (1,2,3,4...)
_FSI_CURRENTDOUBLEPAGE_	1-based index of the current double page (1,2,4,6...) For inner pages this value represents the index of the left page. For the front and back cover the value is the actual page index

9.19 Page Overlays

Page overlays are compiled Flash movies clips that can be used as an overlay, floating above a page in FSI Pages. You have probably seen the "Photo Album" example created with FSI Pages at www.neptunelabs.com. The small sticky photo corners come from an overlay file.

Using Overlays

Overlay files have to reside in the "/plugins/pages/overlays/" directory of your FSI Viewer setup directory. You can add a global overlay to appear on all pages and/or specify an overlay file for specific pages.

Defining a global page overlay

```
<Plugins>
...
  <Plugin src="pages" >
    <PageOverlays>
      <Overlay default="true" file="ov_photo.swf" />
      [...]
    </PageOverlays>
  </Plugin>
...
</Plugins>
```

The example above adds the "sticky photo corners" overlay to all pages, like in the "Photo Album" sample.

Defining overlays for specific pages

```
<Plugins>
...
<PageOverlayShowZoomed value="true" />
<PageOverlays>
  <Overlay default="true"
    file="myDefaultOverlay.swf" />

  <Overlay page="5" file="myOverlay1.swf" />

  <Overlays page="10">
    <Overlay file="myOverlay2.swf"
      x="0.5" y="0.5" />
    <Overlay file="myOverlay3.swf" />
  </Overlays>
</PageOverlays>
...
</Plugins>
```

The example above does the following:

- add a default overlay "myDefaultOverlay.swf"
- adds the overlay file "myOverlay1.swf" to page 5
- adds "myOverlay2.swf" and "myOverlay3.swf" to page 21

The attributes of the <overlay> nodes can be used to specify variables for an overlay.

Note that you can, but don't need to define a default overlay.

9.20 FSI Pages Presets in FSI Server Interface "Publish to Web"

Using FSI Pages with FSI Server Imaging Server you can choose from the following presets when using the "Publish as FSI Pages" option:

- **Catalog with links** Each image represents a page, no page margins, no overlays. If the image contains XML data in the "IPTC_FSI" data field, links will be presented on the page.

- **Catalog without links** Each image represents a page, no page margins, no overlays.
- **Hard cover Book** First and last page of the catalog have a inflexible appearance.
- **Teaser Thumbnail** No user interface, zoom is disabled. Pages flip automatically when idle. You can specify a link for the entire viewer area so that you can use this template to publish a thumbnail of a catalog linking to the full size catalog. See → *Parameters for FSI Pages Thumbnails* for configuration details.
- **Photo Album** A simple photo album with photo corners and an emboss effect for each image.

9.21 Event Notifications and Actions

When using JS Bridge plug-in or FSI Viewer component you can use the following event notifications and actions to interact with FSI Pages Add-on.

Please refer to JS Bridge plug-in for details on how to setup communication between JavaScript and FSI Viewer.

Event ID	Parameter	Description
onPagesInitComplete	Integer	Called after the FSI Pages Add-on is ready for use. The parameter provides the total number of pages.
onPagesGotoPage	Integer	Called each time before the user goes to a different page. The parameter provides the target page number.
onPagesPageChanged	Integer	Called each time after the current page has changed. The parameter provides the index of the page currently visible.
onPagesProgress	Integer	Called each time the load progress of FSI Pages changes. The parameter provides the number of pages remaining in the load queue.
onPagesShowIndex	Boolean	Called each time the user shows (true) or hides (false) the page index.
onPagesShowZoom	Boolean	Called each time the user starts (true) or ends (false) page zoom.
onPagesLinkClick	URL	Called each time the user clicks a link. The parameter contains the URL.

Commands for FSI Pages Add-on

Command	Description
" (empty string)	Stop current action
FirstPage	Go to first page.
LastPage	Go to last page.
NextPage	Go to next page.
PreviousPage	Go to previous page.
GotoPage	<p>Go to page n</p> <p>Important: Use SetVariable("newImageIndex", n) where n is [0 .. pages] prior to sending this command.</p> <p>Alternatively you can specify a page relative to the current page. To go to a page relatively you have to set "newImageIndex" to a number prefixed by "+" or "-", e.g. "+ 1" is the same as "NextPage".</p>
Print	Open or close the print dialog.
Save	Open or close the save dialog.
Search	Open or close the search dialog.
ZoomLeftPage	Zoom the left page
ZoomRightPage	Zoom the right page
NormalPageView	Exit zoom view
ShowHidePageIndex	Show or hide the page index (thumbnail view)

9.22 Parameters to retrieve Image Collections

Server	
Description	Address of imaging server
Syntax	URL
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"

The URL of the setup path of imaging server to be used when retrieving image collections.

Dir	
Description	Image directory
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.0.0, FSI Pages mobile 1.0.0

The directory on imaging server containing the images to use for FSI Pages.

ListTemplate	
Description	Template used to retrieve image list from imaging server
Syntax	String
Default	"fsi/image_list.xml"
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.0.0, FSI Pages mobile 1.0.0

The XML based template to use when retrieving image lists from imaging server via "dir" or "query" parameters. Available templates are located in the "WEB-INF/templates/fsi/" directory of your imaging server setup.

EmptyImages	
Description	Define blank pages
Syntax	String
Default	---
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.3.0 or higher, FSI Pages mobile 1.0.0

Using this parameter you can add blank pages. The value of the parameter must contain one or more image page numbers of the blank pages to be inserted. E.g.

`<EmptyImages value="2,10" />` adds two blank pages at page 2 and page 10, moving the other images in your collection correspondingly. Alternatively you can add `<image empty="true" />` nodes to your image collection.

RemovePages	
Description	Remove individual pages from the collection
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.5.5 or higher, FSI Pages mobile 1.0.0

Using this parameter you can remove images from the image collection. The behavior of FSI Pages is exactly as if the images would not be listed in the image collection. E.g. `<EmptyImages value="1,3,4" />` removes the first, third and fourth image in the collection. Please note that removing images takes place before inserting blank pages using the "EmptyImages" parameter.

ReversePageOrder	
Description	Reverses the order of the individual pages in the collection
Syntax	Boolean
Default	false
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 4.1.5 or higher, FSI Pages mobile 1.0.0

Set this parameter to "true" to reverse the page sequence of pages. Please note that this parameter does not change the reading order. Please see → *Page Reading Order* to change the reading order of a document.

9.23 Basic Parameters

InitialPage	
Description	Page to display on startup
Syntax	Number
Default	1
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.0.0, FSI Pages mobile 1.0.0

Specifies the page to display on startup, e.g. "1" for the front cover.

Note: The parameter "InitialPage" has no effect if you are using the → *RememberLastViewedPage* parameter to make the user to return to the last viewed page. You can use "ForceInitialPage" instead if you want FSI Pages to show a specific page independent of the last viewed page.

PageOrderRTL	
Description	Switch to right-to-left reading order
Syntax	Boolean
Default	False
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 4.1.5 or above, FSI Pages mobile 1.0.0

By default FSI Pages uses the western page reading order (left-to-right). Enable this parameter to use *right-to-left reading order*, e.g. for Arab documents.

FirstPageNumber	
Description	Offset for page numbering
Syntax	Number
Default	1
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.0.0, FSI Pages mobile 1.0.0

By default (FirstPageNumber=1) the front cover is page number one. You can specify an offset so that the page numbers start with a different value. This parameter affects page numbers, the index and the display of the current page number in the FSI Pages user interface.

Skin	
Description	User interface (skin) for FSI Pages
Syntax	String
Default	silver
Context	Pages plug-in node or HTTP query prefixed with "pages_"

Specifies the skin containing the user interface for FSI Pages. Available skins are located in the directory "skins/pages" of your FSI setup directory. Please note that this does not affect the skin of FSI Viewer which can be specified separately using the "skin" parameter of FSI Viewer.

Slider	
Description	Show / hide the page slider
Syntax	Boolean
Default	true
Context	Pages plug-in node or HTTP query prefixed with "pages_"

Show (true) or hide (false) the slider to select pages.

Index	
Description	Enable / disable table of contents
Syntax	Boolean
Default	true
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.0.0, FSI Pages mobile 1.0.0

Enable (true) or disable (false) the user to show the page index containing thumbnails of all pages.

KeepIndex	
Description	Keep index data when hiding the index
Syntax	Boolean
Default	true
Context	Pages plug-in node or HTTP query prefixed with "pages_"

Setting the value to true (default) speeds up showing and hiding the page index with a possible overall performance penalty when presenting many pages. Setting the value to false deletes index data when hiding the index, forcing the viewer to build the index each time it is shown.

PageTurnSpeed	
Description	Speed when turning pages
Syntax	Number
Default	82
Context	Pages plug-in node or HTTP query prefixed with "pages_"

The speed of the page turning effect. "0" is the slowest and "100" the fastest value possible.

PageTurnAnimation	
Description	Animation when flipping to another page
Syntax	String
Default	"flip"
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	6

Defines the animation when flipping to another page. "flip" is the default and shows a page turn effect. "slide" slides the pages as if they were positioned next to each other (like in FSI Pages mobile).

Zoom	
Description	Enable / disable zooming into pages
Syntax	Boolean
Default	true
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.0.0, FSI Pages mobile 1.0.0

Enable (true) or disable (false) zooming into pages with FSI Viewer when clicking on a page.

ShowLoadProgress	
Description	Show / hide the load progress display
Syntax	Boolean
Default	true
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.1.5 or above

Show (true) or hide (false) the load progress display. FSI Pages automatically sets this parameter to "true" if you specify a hyperlink for the entire FSI Pages instance (see → *DemoURL*).

AutoCropPages	
Description	Crop images to fill entire page
Syntax	Boolean
Default	false
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	3.5.0 or higher, FSI Pages mobile 1.0.0

Enable (true) this option to automatically crop images to fill the entire pages. You need to specify an aspect ratio manually in this case.

ClickPageToZoom	
Description	Enable / Disable zooming by clicking a page
Syntax	Boolean
Default	true
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.1.5 or higher

Enabling (true) this option enables the user to zoom a page by clicking it.

ZoomAreaAnimationSpeedIn	
Description	Defines the animation speed of in place magnifications initiated by "#zoomarea" hyperlinks
Syntax	Number
Default	520
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	5.0.0 or above

Defines the speed of the zoom animation when zooming **in**.

ZoomAreaAnimationSpeedOut	
Description	Defines the animation speed of in place magnifications initiated by "#zoomarea" hyperlinks
Syntax	Number
Default	240
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	5.0.0 or above

Defines the speed of the zoom animation when zooming **out**.

ZoomAreaAnimationType	
Description	Defines the animation type of in place magnifications initiated by "#zoomarea" hyperlinks
Syntax	Number
Default	5
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	5.0.0 or above

Defines the animation type of the zoom animation.

ZoomAreaTargetScale	
Description	Magnification level of in place magnifications initiated by "#zoomarea" hyperlinks
Syntax	Number
Default	200
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	5.0.0 or above

Defines the zoom factor for the link. Two limitations apply: The zoomfactor must be 110 or greater, otherwise no zoom area is displayed. The zoomarea may not exceed the FSI Pages instance size.

DoublePageAllowInteractivity	
Description	Enable or disable interactivity in double page zoom mode
Syntax	Boolean
Default	false
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	5.0.0

When enabled the image will scroll when the user moves the mouse pointer.

DoublePageZoom	
Description	Enables the interactive zoom mode
Syntax	Boolean
Default	false
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	5.0.0

If enabled the zoom mode will display both pages at the given zoom level. When disabled, interactivity (like clicking hyperlinks) is disabled in double page zoom mode.

DoublePageZoomScale	
Description	Defines the zoom level of double page zoom
Syntax	Number
Default	1:1
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	5.0.0

Default: "1:1" (zoom to original size)

a) relative to source image resolution ("1:1", "1.5:1", "1:2?...")

b) relative to current page size in percent (200, 300, ...)

DoublePageZoomDragPan	
Description	enables dragging of the image while being in double page zoom
Syntax	Boolean
Default	true
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	6

With this option enabled you can drag the image while being in double page zoom mode. Otherwise, the double page zoom moves according to the mouse position.

ForceXMLDataFiles	
Description	Force XML data useage
Default	false
Context	Pages plug-in
Version	FSI Pages 4.1.5 or higher, FSI Pages mobile 1.0.0

If set to true, the IPTC data from the image file is ignored and the links and overlay information is retrieved from the XML files. If set to false, XML data is used only if no IPTC data is present in the image.

9.24 Parameters for the Front- and Backcover

FrontCover	
Description	Enable/disable the front cover
Syntax	Boolean
Default	true
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.5.5 or higher, FSI Pages mobile 1.0.0

Setting "FrontCover" parameter to false disables access to the front cover and forces FSI Pages to display the first image on the first inner left page.

BackCover	
Description	Enable/disable the back cover
Syntax	Boolean
Default	true
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	3.5.5 or higher

Setting "BackCover" parameter to false disables access to the back cover and forces FSI Pages to add a blank content page if required.

BlankBackCover and BlankFrontCover	
Description	Use a blank page for front and/or back cover
Syntax	Boolean
Default	false
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.0.0, FSI Pages mobile 1.0.0

Use a blank page as front and/or back cover. The cover page(s) will be added to images in the image list.

FullBackCover and FullFrontCover	
Description	Fill entire front and/or back cover with image
Syntax	Boolean
Default	false
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.0.0, FSI Pages mobile 1.0.0

Fill the entire front and/or back cover with the corresponding image. This is: Ignore page margins and do not use an overlay for the cover page specified.

BackCoverImage and FrontCoverImage	
Description	Use the specified image for front or back cover
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"

A path to an image to use for the front or back cover. FSI Pages will use the → *FSIBase* parameter of FSI Viewer for relative paths. The cover page(s) will be added to the images in the image list.

BackCoverConfig and FrontCoverConfig	
Description	Use the specified image configuration file for front or back cover
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"

Specify a separate image configuration file (*.xml) to be used for the front or back cover respectively. The cover page(s) will be added to images in the image list.

HardPages	
Description	Defines page flexibility
Syntax	Boolean
Default	false / all / cover
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	5.0.0

Defines wether the pages are flexible or inflexible e.g. like a book cover.

PageBackgroundImage	
Description	Relative path to the image on your imaging server
Syntax	Filename
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	5.0.0

The path to an image on your imaging server to be used as a background. If you use a relative path (recommended) the → *FSIBase* parameter will be used to get the absolute path to the image.

PageBackgroundImageEffects	
Description	Image effects for the background image
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	5.0.0

Defines image modification parameters to apply to images used. The value of this parameter depends on the imaging server being used.

With FSI Server you can sharpen the image and define the JPEG compression level, e.g. "effects=sharpen(230)&quality=95".

9.25 Parameters defining Layout and Appearance

Ratio	
Description	Aspect ratio of pages
Syntax	String
Default	"auto"
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.0.0, FSI Pages mobile 1.0.0

The aspect ratio of pages displayed in FSI Pages.

You can specify any aspect ration (e.g. "320:240" or "1:2") or specify "auto" to use the aspect ratio of the first image in the image collection.

SingleSided	
Description	Use images on uneven or even pages only
Syntax	String
Default	false / even / uneven
Context	Pages plug-in node or HTTP query prefixed with "pages_"

By default FSI Pages creates a page for each image in the image collection.

Using this parameter you can advise FSI Pages to use images on even or uneven pages only, inserting a blank page on the opposite side.

Use "false" to display images on all pages. (default).
 Use "even" to display images on even pages only.
 Use "uneven" to display images on uneven pages only.

BendEffect	
Description	Visual "3D" of the pages
Syntax	"Glossy" or "Matte"
Default	"Glossy"
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.0.0 or above

The effect "Glossy" adds highlights and shadows, while "Matte" adds shadows only (like with FSI Pages version 3). Please use → *BendEffectIntensity* parameter to modify the effect level or disable the visual effect.

BendEffectIntensity	
Description	Level of visual "3D" effects
Syntax	Number
Default	65
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.0.0 or above

Defines the 3D effect level of the pages in percent. Using a value of "0" disables the visual 3D effect. The value additionally affects the opacity of the drop shadow.

DropShadow	
Description	Enable or diable the dropshadow effect for the pages
Syntax	Boolean
Default	true
Context	Pages plug-in node or HTTP query prefixed with "pages_ "
Version	4.0.0 or above

Enables or disables the drop shadow of the pages.

DropShadowDistance	
Description	Distance of the page's drop shadow
Syntax	Number
Default	50
Context	Pages plug-in node or HTTP query prefixed with "pages_ "
Version	4.0.0 or above

Distance of the pages' drop shadow in percent, where "100" is the greatest possible distance.

ImageAlign	
Description	Alignment of images on the pages
Syntax	String
Default	"TC"
Context	Pages plug-in node or HTTP query prefixed with "pages_ "
Version	FSI Pages 3.0.0, FSI Pages mobile 1.0.0

Alignment of the images on the pages. The default is "TC" which centers the images horizontally on top of the page.

Please specify the alignment for even (left) pages. Uneven (right) pages will be aligned correspondingly. Possible values are any combination of:

T (Top) and L (Left)
 C (Center) and C (Center)
 B (Bottom) and R (Right)

PageColor	
Description	Page background color
Syntax	HexColor
Default	FFFFFF
Context	Pages plug-in node or HTTP query prefixed with "pages_ "
Version	FSI Pages 3.0.0, FSI Pages mobile 1.0.0

Background color of the pages specified as 6 digit web color.

PageBorderColor	
Description	Color of page borders
Syntax	HexColor
Default	CCCCCC
Context	Pages plug-in node or HTTP query prefixed with "pages_ "
Version	FSI Pages 3.0.0, FSI Pages mobile 1.0.0

Color of the page borders specified as 6 digit web color. Use "false" for no border at all.

HelpImageURL	
Description	Help image to be displayed left of the first page
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_ "
Version	3.6.0 or above

An absolute or relative ImageURL (like FPXSrc) to an image on the imaging server that FSI Pages displays left of the first page of the catalog.

ScrollBaseColor	
Description	Base color for the scroll bar
Syntax	HexColor
Default	CCCCCC
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	3.5.0 or higher

6-digit hex-color for the scroll bar in the page index.

ScrollArrow	
Description	Color for the scroll bar's arrows
Syntax	HexColor
Default	000000
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	3.5.0 or higher

6-digit hex-color for the arrows of the scroll bar in the page index.

ScrollTrack	
Description	Color for the scroll bar's track bar
Syntax	HexColor
Default	DDDDDD
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	3.5.0 or higher

6-digit hex-color for the track bar of the scroll bar in the page index.

MarginBottom, MarginLeft, MarginRight, MarginTop	
Description	Margin of the turning pages
Syntax	Number in pixel
Default	0
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.0.0, FSI Pages mobile 1.0.0

Margin on each side of the turning pages in pixel. The margin effects the position of the pages only, not the user interface of FSI Pages. You should at least specify a bottom margin to provide space for the interface.

PageMarginBottom, PageMarginInner, PageMarginOuter, PageMarginTop	
Description	Margin of the page content
Syntax	Number in percent of the page width / height
Default	0
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.0.0, FSI Pages mobile 1.0.0

Margins of the content of each page in percent of the page width (inner and outer) or page height (top and bottom).

ViewerMarginBottom, ViewerMarginLeft, ViewerMarginRight, ViewerMarginTop	
Description	Margin of FSI Viewer
Syntax	Number in pixel
Default	0
Context	Pages plug-in node or HTTP query prefixed with "pages_"

Margin on each side of FSI Viewer in pixel when zooming into a page. Usually you need to set "ViewerMarginBottom" to the height of the FSI Pages interface.

ThumbSize	
Description	Max. size of thumbnails in the page index
Syntax	Number in pixel
Default	60
Context	Pages plug-in node or HTTP query prefixed with "pages_"

The maximum size of thumbnails displayed in the index. The actual width and height depend on the aspect ratio of the pages (see parameter → *Ratio*).

ThumbPadding	
Description	Padding between thumbnails in the index
Syntax	Number in pixel
Default	10
Context	Pages plug-in node or HTTP query prefixed with "pages_"

The padding between thumbnails displayed in the page index in pixel.

PageNumbers	
Description	Display page numbers on pages
Syntax	Boolean
Default	true
Context	Pages plug-in node or HTTP query prefixed with "pages_"

Show or hide page numbers on pages.

PageNumPos	
Description	Alignment of page numbers
Syntax	String
Default	"BL"
Context	Pages plug-in node or HTTP query prefixed with "pages_"

PageNumBGColor	
Description	Background color of page numbers
Syntax	HexColor
Default	false
Context	Pages plug-in node or HTTP query prefixed with "pages_"

6-digit hex color of page numbers or "false" for no background.

UseRomanPageNumbersToPage	
Description	Display Roman page numbers up to page number n
Syntax	Number
Default	0
Context	FSI Pages
Version	5.0.4, FSI Pages mobile 1.0.0

Defines the page number up to which FSI Pages uses Roman page Numbers. The default value of "0" forces FSI Pages to display Roman numbers for negative numbers and 0.

If you set the value to "5" the display will be like this:
I,II,III,IV,V,6,7,8,9...

Note that you can use the Parameter → *FirstPageNumber* to offset all page number values.

RomanPageNumbersOffset	
Description	Offset Roman page numbers by given value
Syntax	Number
Default	0
Context	FSI Pages
Version	FSI Pages 5.0.4, FSI Pages mobile 1.0.0

Defines the value of the first Roman page number.

The default value of "0" means no offset so that the Roman numbers start with "I".

CustomPageNumbers	
Description	Comma separated string of custom page numbers
Syntax	Number
Default	_____
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 5.5.5 or higher, FSI Pages mobile 1.0.0

Use "CustomPageNumbers" to specify custom page numbers as a comma separated string. Items starting with underscore ("_") will not be displayed. Pages without a custom page number will use the ordinary page numbering.

Example: <CustomPageNumbers value="_cover1,_cover2,3,4,5,_" />

In this case the cover page and the first left page will have no page number. The first right page will start with page number "3". The sixth page again has no page number and the seventh and following page will be numbered as usual.

Note: The parameter → *FirstPageNumber* has no impact on the custom page numbers. You can therefore use this parameter to modify the value of the first ordinary page number.

CustomPageNumbersFile	
Description	URL or path to an XML file containing custom page numbers
Syntax	String
Default	_____
Context	plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 5.5.5 or higher

Same as CustomPageNumbers, but the page number values will be loaded from an XML file. You can either define a path relative to the fsi/config/pages_xml/ directory or an absolute URL. The required format of the XML file is as follows:

```
<fsi>
<pagenumbers>3, 6, 7</pagenumbers>
</fsi>
```


9.26 Parameters for Links on Pages

FollowLinks	
Description	Enable or disable FSI Pages to open links
Syntax	Boolean
Default	true
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.6.0 or above, FSI Pages mobile 1.0.0

Setting this parameter to "false" prevents FSI Pages from opening links on the pages. Links and tool tips will nevertheless be displayed.

ShowLinksWhileFlipping	
Description	Show or hide links during flip animations
Syntax	Boolean
Default	false
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.0.0 or above, FSI Pages mobile 1.0.0

Set this parameter to "true" if you want FSI Pages to display link shapes during flip animations. Showing links decreases the performance of the flip animations.

LinkFillAlpha	
Description	Opacity of the link area
Syntax	Number
Default	0
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	3.5.0 or higher, FSI Pages mobile 1.0.0

Specify the opacity of the link areas from 0 (no fill) to 100 (opaque). The color corresponds to the color of the link area's border color.

LinkRGBANormal	
Description	Color and opacity of links in normal state
Syntax	RGBA
Default	0000FFFF
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.0.0, FSI Pages mobile 1.0.0

8-digit hexadecimal number specifying the color and opacity of links on the pages in normal state in the form "RRGGBBAA". E.g. "FF0000FF" for opaque red or "00FF0099" for semi-transparent green

LinkRGBAMHover	
Description	Color and opacity of links in hover state
Syntax	RGBA
Default	FF00FFFF
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.0.0, FSI Pages mobile 1.0.0

8-digit hexadecimal number specifying the color and opacity of links on the pages when the user points at a link in the form "RRGGBBAA". E.g. "FF0000FF" for opaque red or "00FF0099" for semi-transparent green

LinkRGBAAActive	
Description	Color and opacity of links in active state
Syntax	RGBA
Default	FF0000FF
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.0.0, FSI Pages mobile 1.0.0

8-digit hexadecimal number specifying the color and opacity of links on the pages when the user clicks a link in the form "RRGGBBAA". E.g. "FF0000FF" for opaque red or "00FF0099" for semi-transparent green.

ForceLinkURL	
Description	Force URL for all links on pages
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	3.6.0 or above, FSI Pages mobile 1.0.0

Force FSI Pages to use this URL for all links on all pages. This overwrites any URLs specified in the page data.

ForceLinkTarget	
Description	Force a target frame for all hyperlinks on all pages
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	3.6.0 or above, FSI Pages mobile 1.0.0

Force FSI Pages to use this HTML target frame for all links on all pages. This overwrites any target frames specified in the page XML data.
Note: this parameter does not affect "javascript:" and "mailto:" links.

ForceJavascriptTarget	
Description	Force a target frame for all "javascript:" links on pages
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 4.1.1 or above, FSI Pages mobile 1.0.0

Force FSI Pages to use this HTML target frame for all "javascript:" links on all pages. This overwrites any target frames specified in the page XML data.

Note: this parameter does not affect URL and "mailto:" links.

ForceLinkUriPrefix, ForceLinkUriSuffix	
Description	Force a link prefix / suffix for all links on pages
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.6.0 or above, FSI Pages mobile 1.0.0

Force FSI Pages to use this prefix/suffix for all links on all pages. This overwrites any prefixes and suffixes specified in the page data.

Prefixes and suffixes will not be applied to absolute URLs and → *Special URL Values* like "javascript:".

ForceLinkTip	
Description	Force a tool tip for all links on pages
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.6.0 or above, FSI Pages mobile 1.0.0

Force FSI Pages to use this tool tip for all links on the pages. This overwrites any tool tips specified in the page data.

DefaultLinkURL	
Description	Default URL for links on pages
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.6.0 or above, FSI Pages mobile 1.0.0

Use this URL for links if no URL has been specified in the page data.

DefaultLinkTarget	
Description	Default target frame for all hyperlinks on all pages
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.6.0 or above, FSI Pages mobile 1.0.0

Use this HTML target frame for links if no target frame has been specified in the page XML data.

DefaultJavaScriptTarget	
Description	Default target frame for all "javascript:" hyperlinks on all pages
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 4.1.1 or above, FSI Pages mobile 1.0.0

Use this HTML target frame for "javascript:" links if no target frame has been specified in the page data.

DefaultLinkUrlPrefix, DefaultLinkUrlSuffix	
Description	Default prefix / suffix for link URLs on pages
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.6.0 or above, FSI Pages mobile 1.0.0

Use this prefix or suffix for all links on the pages if no prefix / suffix has been specified in the page data.

DefaultLinkTip	
Description	Default tool tip for links on pages
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.6.0 or above, FSI Pages mobile 1.0.0

Use this tool tip for all links on the pages if no tool tip has been specified in the page data.

LinkTemplates	
Description	Template identifiers for link URLs
Syntax	String(s)
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.0.0, FSI Pages mobile 1.0.0

You can provide one or more identifiers that will be replaced in Link URLs. The identifiers are case sensitive and must be separated by commas. The values to insert have to be defined by the parameter "LinkTemplateData" explained below. Please refer to the chapter → *9.18 Modifying Link Values at Runtime* above for details.

LinkTemplateData	
Description	Values to replace template identifiers in link URLs
Syntax	String(s)
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.0.0, FSI Pages mobile 1.0.0

When defining `→ LinkTemplates` to replace place holders in Link URLs you can use this parameter to specify the values that shall be inserted. The number and sequence of the values must match the number and sequence of the templates you defined using the "LinkTemplates" parameter. Multiple values have to be separated by commas and the individual values have to be provided url-encoded (see `→ Escaping or URL-encoding parameter values` Please refer to the chapter `→ Modifying Link Values at Runtime` above for details.

9.27 Parameters for FSI Pages Thumbnails

Due to the Single Source Imaging concept you can display a small, flipping thumbnail of your catalog just by using a different configuration file and choosing a smaller dimension for the FSI Pages instance. The parameters below focus on thumbnail pages.

Menu	
Description	Show / hide the user interface
Syntax	Bool
Default	true
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.0.0, FSI Pages mobile 1.0.0

Show (true) or hide (false) the user interface of FSI Pages add-on.

IdleAutoTurn	
Description	Turn pages automatically
Syntax	Boolean
Default	false
Context:	Pages plug-in node or HTTP query prefixed with "pages_"

Run the FSI Pages in demonstration mode and turn pages automatically when the user does not interact (move the mouse) for n seconds, as defined by "IdleAutoTurnDelay"

IdleAutoTurnDelay	
Description	Min. delay between automatic page flips
Syntax	Floating Point Number (seconds)
Default	2
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	3.6.0

Show each page for at least n seconds when flipping pages automatically. Please note that a page might be displayed for longer if loading the page takes longer than the time specified.

IdleAutoTurnMaxPage, IdleAutoTurnMinPage	
Description	Auto flip range
Syntax	Number (page)
Default	(all pages)
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	3.6.0

Loop the auto flip action between IdleAutoTurnMinPage and IdleAutoTurnMaxPage. This way you can restrict the auto flip to a range of pages. If the initial page is outside the range specified, FSI Pages flips towards the range before looping between the range specified.

DemoURL	
Description	Global link when clicking FSI Pages
Syntax	URL
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	3.5.5 or higher

Specifies a link to open when the user clicks anywhere on the FSI Pages instance. This parameter is especially useful when using a small preview of a catalog to lead the user to the full size version of FSI Pages. You can use URLs starting with "javascript:" to have a Javascript function called when the user clicks the viewer.

You can also use the parameter value "#fullscreen" for the "DemoURL" parameter. Doing so FSI Viewer will enter full screen mode when clicking the FSI Viewer instance.

DemoURLTarget	
Description	Global link when clicking FSI Pages
Syntax	URL
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	3.5.5 or higher

HTML frame in which to open the URL specified by "DemoURL" parameter in. You can use default frame names like "_self" or "_blank" or use a custom frame of your HTML frame set.

Rotation	
Description	Rotation in degrees of the thumbnail
Syntax	Number
Default	0
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.1.0 or higher

Optional rotation of FSI Pages thumbnails. Please note that using the "rotation" parameters with normal size FSI Pages instances is not recommended as zooming, printing, index view and alike do not correspond to the rotation.

PageInput	
Description	Show or hide Page Input field
Syntax	Boolean
Default	true
Context	Pages plug-in
Version	4.0.0 or higher

Defines whether the input form for page numbers is shown.

ButtonsFirstLastPage	
Description	Show or hide First/Last Page buttons
Syntax	Boolean
Default	true
Context	Pages plug-in
Version	4.0.0 or higher

Defines whether the navigation buttons for jumping to the first or last page are shown.

9.28 Parameters for Statistics

(Requires FSI Viewer 4.1.3 or above)

The following parameters can be used to measure the usage of FSI Pages instances by means of third party statistic tools. Using these parameters you can define a template based URL to be called each time the user performs a specified action like flipping to another page. The templates in the URL will be replaced at runtime. You can specify a custom delay before the statistics URL will be loaded.

PageLogURL	
Description	Template based URL for page view logging
Syntax	URL
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.1.3 or higher

Template based URL to load each time the user flips to another page.
The following templates will be replaced at runtime:

- [uniqueid] The FSI Pages parameter value "UniqueID"
- [instanceid] A unique number identifying the FSI Pages instance (unique per instance)
- [pagenum] The page number of the current left page
- [targetpage] The target page number to flip to (if applicable)
- [totaltime] Total seconds the FSI Pages instance is running
- [action] String identifying the action which caused the log event (one of: flip, chapters, index, input, hyperlink, hyperlinkurl)
- [hyperlinkurl] The URL of the clicked hyperlink (if applicable)

Example URL:

```
http://example.com/log.php?catalog=foo&action= [action] &page= [pagenum]
&targetpage= [targetpage] &uniqueid= [uniqueid] &totaltime= [totaltime] &IID=
[instanceid]&targeturl=[hyperlinkurl]
```

You can alternatively call a Javascript function using the "javascript:" syntax. In this case you need to define a valid HTML target frame using the "PageLogTarget" parameter.

PageLogEvents	
Description	Send PageLog Event Parameters
Syntax	String
Default	flip
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	5.5.3

This defines which parameters are sent to the PageLogURL when using the pagelog feature of FSI Pages.

flip	flipping a page
chapters	selecting an entry in the drop down
index	selecting a page from the thumbnail page index
input	goto page from page number input
hyperlink	flip to a page via "#page" or "#zoomonpage" hyperlink
hyperlinkurl	clicking a hyperlink that links to an URL, Javascript call etc.
zoom	[targetpage] contains the index of the zoomed page
print	[targetpage] contains the index of the saved page
save	[tagretpage] contains the index of the saved page

PageLogTarget	
Description	HTML Target Frame to open the PageLogURL in
Syntax	String
Default	_internal
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.1.3 or higher

By default FSI Pages uses the special target "_internal" which loads the URL from within the Flash plug-in. You can use this parameter to specify a different HTML target frame like "_self" or "_top".

PageLogDelay	
Description	Page View logging delay
Syntax	Number (milliseconds)
Default	2000
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.1.3 or higher

Specifies a minimum period of time a page must be visible before loading the PageLogURL.

Use "0" to log each page access immediately. The default value of 2000 (ms) ensures, that a page has been viewed for at least 2 seconds.

9.29 Advanced Parameters

RememberLastViewedPage	
Description	Store the most recently viewed page locally
Syntax	Boolean
Default	false
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.0.5 or above

If you enable this parameter FSI Pages stores the page number of the most recently viewed page on the user's computer and displays this page on start up the next time the user views the same catalogue (image collection) again.

Note: The parameter → *InitialPage* has no effect if there is a valid value for the last viewed page. You can use "ForceInitialPage" if you want FSI Pages to show a specific page independent of the last viewed page.

Note: When using image collections retrieved from FSI Server, FSI Pages uses the → *Dir* parameter to identify the image collection automatically. If you create image collections manually, you additionally need to add the parameter → *UniqueID* so that the "last viewed page" can be assigned to this image collection.

RememberLastViewedPageExpireAfter	
Description	Expire time for "RememberLastViewedPage" parameter in seconds
Syntax:	Number (seconds)
Default	Undefined (infinite)
Context:	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.0.9 or above

By default the last viewed page will be restored on the next visit of a user if you enable the RememberLastViewedPage parameter. You might want to specify an expire time after that the last viewed page will be discarded. Please specify the expire time in seconds, e.g. "3600" for one hour.

AutoZoom	
Description	Initial magnification in FSI Viewer
Syntax	Number in percent of the source image or "fitWidth" or "fitHeight"
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"

The initial magnification in FSI Viewer after the user selected an image to zoom. Specify the magnification in percent of the source image, e.g. "100" zooms to the physical resolution of the source image (default maximum magnification). With FSI Viewer 4.1 or above you can additionally use the parameter values "fitWidth" or "fitHeight" to make FSI Viewer zoom a page to fit the entire width or height of a page. If you do not specify this parameter FSI Viewer displays the entire image.

PreloadForward	
Description	Number of pages to pre-cache
Syntax	Number
Default	4
Context	Pages plug-in node or HTTP query prefixed with "pages_"

Number of subsequent pages to load in advance.

PreloadBackward	
Description	Number of pages to pre-cache
Syntax	Number
Default	2
Context	Pages plug-in node or HTTP query prefixed with "pages_ "

Number of preceding pages to load in advance.

PageCacheLimit	
Description	Number of pages to hold in memory
Syntax	Number
Default	8
Context	Pages plug-in node or HTTP query prefixed with "pages_ "
Version	4.1.3

Defines the number of pages to hold in memory. Using larger numbers will make FSI Pages consume more memory (RAM) and cached pages will be read from the browser's file cache less frequently. Use "false" to hold all pages in memory once loaded. This is the default behaviour of previous versions.

PageCacheRemoveTime	
Description	Minimum time each page is being held in memory
Syntax	Number (seconds)
Default	3
Context	Pages plug-in node or HTTP query prefixed with "pages_ "
Version	4.1.3

Defines the minimum time each page is being held in memory after it has been viewed the last time. Set this parameter to "0" to remove pages exceeding the "PageCacheLimit" immediately.

PageOverlays	
Description	Defining overlay(s) for all or a specific page
Syntax	XML
Default	—
Context	Pages plug-in node
Version	4.0.0 or above

Define a global page overlay for all pages and/or for individual pages. A page overlay is a movie clip that loads on top of a page. To specify a default overlay for all pages except for those, you specified an individual overlay for, please add a `default="true"` attribute to the `<overlay>` node. Please see chapter → 9.19 *Page Overlays* for details.

Defining a global page overlay

```

<Plugins>
...
  <Plugin src="pages" >
    <PageOverlays>
      ...
        <Overlay default="true"
          file="ov_photo.swf" />
        <Overlay page="5"
          file="myOverlay1.swf"
          posX="6" posY="12"/>
        <Overlay page="21"
          file="myOverlay2.swf"
          posX="0" posY="98"/>
      ...
    </PageOverlays>
  </Plugin>
...
</Plugins>

```

The example defines the default overlay "ov_photo-swf" and defines individual overlays for pages 5 and 21.

PageOverlayShowZoomed	
Description	Show overlays when zooming pages
Syntax	Boolean
Default	false
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.0.0 or above

Defines whether to show or hide page overlays when zooming into a page.

MovePages	
Description	Move pages when zooming
Syntax	Boolean
Default	true
Context	Pages plug-in node or HTTP query prefixed with "pages_"

Move the pages when zooming to maximize the space available for FSI Viewer. If set to "false" images will be zoomed in-place.

XMLBase	
Description	Path to XML page data
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 3.0.0, FSI Pages mobile 1.0.0

An absolute or relative URL base to XML files containing data like hyperlinks for FSI Pages. If the IPTC data of the images does not contain data for FSI Pages, the viewer will load XML data from the location specified. Alternatively you can force loading XML files rather than using IPTC data by setting the `ForceXMLDataFiles` parameter to "true".

The complete URL to load the XML file from will be created based on the XMLBase parameter value plus the image path with an ".xml" extension.
Example:

"http://example.com/xmldata" [^] + "/images/catalog/page_001.tif" + ".xml"

You can set the XMLBaseUseImageFileName to "true" to use the image file name instead of the entire image path.

Example:

"http://example.com/xmldata" [^] + "/page_001.tif" + ".xml"

If you specify this parameter, FSI Pages will try to load XML data for each page from the location specified. The file path of the XML files must match XMLBase/[image filename].xml

XMLBaseUseImageFileName	
Description	Use file name or path to build XML data URLs
Syntax	Boolean
Default	false
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	FSI Pages 5.6.3

By default the image path will be added to the XMLBase to build the URL to XML data files. Set the parameter to true to use file names instead. In this case, you will need to specify XMLBase differently for each publication.

Print	
Description	Enable printing of pages
Syntax	Boolean
Default	false
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	3.5.0 or higher

Enable or disable the user to print the content of the current page. Please refer to the chapter → *Printing Pages* for details.

PrintResolution	
Description	Max. size of the image to print
Syntax	Number (pixels)
Default	2000
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	3.6.0 or higher

Defines the maximum size of the image in pixels to download for printing. Please note that the actual size depends on the size of the image defined in the real time template used with FSI Viewer (see parameter → *PrintTemplate*).

PrintOptions	
Description	Options to show in the print dialog
Syntax	String
Default	"range,both,left,right,view"
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	6

Allows you to change the options which are displayed in the print dialog. By default, all options available (Range, Both, Left, Right, View) are shown. Please refer to the chapter → *Printing Pages* for details.

PrintEffects	
Description	Image effects for the image to print
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	3.6.0 or higher

Defines image modification parameters to apply to images used for printing. The value of this parameter depends on the imaging server being used. With FSI Server you can sharpen the image and define the JPEG compression level, e.g. "effects=sharpen(230)&quality=95".

PrintTemplate	
Description:	Real time template for the image to print
Syntax	String
Default	fsi
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version:	3.6.0 or higher

Defines the FSI Server profile used to load the print image. By default FSI Pages uses the default real time template "fsi".

Save	
Description	Enable / disable saving of pages
Syntax	Boolean
Default	false
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	3.6.0 or higher

Enables save functionality. Using the save function you can enable the user to open or download a document related to the current page – e.g. the source PDF document of a catalog. Please refer to the chapter → *Saving Pages* for details.

SaveDocumentFile	
Description	URL to a PDF document containing all pages
Syntax	URL
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.0.0 or higher

URL to a document containing all pages displayed in an FSI Pages instance. This is the URL the user will be redirected to when choosing the "Entire Document" option in the "Save" dialog. Please refer to the chapter → *Saving Pages* for details.

SaveURLPrefix and SaveUriSuffix	
Description	Prefix or suffix for page download URLs
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	3.6.0 or higher

Specifies a prefix and/or suffix for the URL FSI Pages opens when the user chooses the "Left Page" or "Right Page" option in the Save dialog. Please refer to the chapter → *Saving Pages* for details.

SaveURLTarget	
Description	Target frame for downloading a file related to a page
Syntax	String
Default	_blank
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	3.6.0 or higher

HTML frame to open page related files (SaveURL) in. Please refer to the chapter → *Saving Pages* for details.

ForceSaveURL	
Description	URL to use for downloading a file related to a page
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	3.6.0 or higher

This parameter overrides the download URL specified in the "file" attribute of the page's XML data. Specifying this parameter forces FSI Pages to use the same download URL for all pages. Please refer to the chapter → *Saving Pages* for details.

SaveAllowSavingImages	
Description	Add image download options to the "Save" dialog
Syntax	Boolean
Default	true
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.0.0 or higher

Set this parameter to "false" if you want to prevent the user from downloading images of pages. In this case the PDF download options will be displayed only. Please refer to the chapter → *Saving Pages* for details.

SaveResolution	
Description	Max. resolution for downloading images
Syntax	Number (pixel)
Default	2000
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.0.0 or higher

The maximum resolution for images the user can choose to download in the "Save" dialog. Please note that the actual size of the image might be limited by the resolution of the source image or by an imaging server restriction. Please refer to the chapter → *Saving Pages* for details.

SaveEffects	
Description	Image effects for the image to download
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.0.0 or higher

Defines image modification parameters to apply to images used for downloading. The value of this parameter depends on the imaging server being used. With FSI Server you can sharpen the image and define the JPEG compression level, e.g. "effects=sharpen(230)&quality=95".

SaveTemplate	
Description	FSI Server profile to use when downloading images
Syntax	String
Default	fsi
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.0.0 or higher

The FSI Server profile to use when downloading images in the "Save" dialog. Please refer to the chapter → *Saving Pages* for details.

SaveOptions	
Description	Enable or disable options in the "Save" dialog
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.0.0 or higher

A comma separated list of options to be presented to the user in the "Save" dialog. The default is presenting all options which corresponds to the following parameter value:

document,chapters,both,left,right

Using the parameter value: left,right enables saving the left or right page only. Please note that the options might as well not appear, if there is no data available or specified to download for any of these options. Please refer to the chapter → *Saving Pages* for details.

Search	
Description	Enable searching a catalog
Syntax	Boolean
Default	false
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.0.0 or higher

Enables or disables the search interface of FSI Pages. Please refer to the chapter → *Searching in FSI Pages* for details.

SearchType	
Description	Use "simplesearch" or "search" type for FSI Server
Syntax	String
Default	"simplesearch"
Context	plug-in node or HTTP query prefixed with "pages_"
Version	5.5.6 or higher (FSI Server only)

Using "simplesearch" makes FSI Server search in the data fields "iptc.fsi_search_data" and "iptc.caption" only by default rather than searching in all meta data fields. When using FSI Server 1 please set the value of this parameter to "search" in your _default.xml

SearchContext	
Description	meta data fields to search
Syntax	String
Default	"iptc.fsi_search_data,iptc.caption, iptc.fsi_extra"
Context	plug-in node or HTTP query prefixed with "pages_"
Version	5.5.6 or higher (FSI Server only)

Comma separated list of meta data fields names to search in.
FSI Server with FSI Viewer parameter SearchType="simplesearch" only.

SearchDialogHeight and SearchDialogWidth	
Description	Size of the search dialog
Syntax	Number (pixel)
Default	400 (width) / 300 (height)
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.0.0 or higher

The size of the search dialog in pixel. Please refer to the chapter → *Searching in FSI Pages* for details.

SearchAutoWildCards	
Description	Add wildcards ("*") to keywords automatically
Syntax	Boolean
Default	true
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.0.0 or higher (depricated for FSI Server)

By default FSI Pages adds wildcards ("*") to the keywords the user entered into the search input of the search dialog. Set this parameter to "false" if you don't want FSI Pages to add wildcards automatically. Please refer to the chapter → *Searching in FSI Pages* for details.

SearchAutoSelectFirstResult	
Description	Search Complete Action
Syntax	Boolean
Default	true
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	5.5.4

Select the first search result item when the search is complete (true, default) or just show the results (false).

SearchResultClickAction	
Description	Defines action when search result is clicked
Syntax	Enum
Default	flip
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	5.5.4

flip: flip to result page (default)

zoom: zoom the result page

SearchTextOnStart	
Description	Search text on startup
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	5.5.4

Define a text to search for on start or an empty string (default) to not initiate a text search on start.

SearchAutoExactMatch	
Description	Add quotes (") to keywords automatically
Syntax	Boolean
Default	false
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.1.5 or higher

Setting this parameter to "true" makes FSI Pages add quotes to search requests so that only exact matches will be returned. Setting this parameter implicitly disables the → *SearchAutoWildCards* parameter.

SearchParameters	
Description	Adds parameters (modifiers) to the search query
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.1.5 or higher

This parameter provides the possibility to add parameters to the search query within FSI Pages. The parameters need to be provided in the regular HTTP request format: param1=value1¶m2=value2[...].

SearchCustomURL	
Description	URL to request the search result from
Syntax	URL
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.0.0 or higher

You can assign a custom URL to direct search requests to using this parameter. For example you might want to use a database and server side script to return pages related to the search request. This is the default way if you are not using imaging server. Please refer to the chapter → *Searching in FSI Pages* for details.

SearchCustomValue	
Description	A custom value to pass with the search request
Syntax	String
Default	—
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.0.0 or higher

A custom string to pass to the server along with the search request. This option is useful when using a server other than FSI Server to return search results. Please refer to the chapter → *Searching in FSI Pages* for details.

SearchQueryTemplate	
Description	Template for the search value
Syntax	String
Default	See description below
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	5.5.0 or higher

Specifies the template to create the search query value.

When using FSI Server the default template is:

```
iptc.caption:%%searchtext%%%20iptc.fsi_search_data:%%searchtext%%
```

This limits the search result to matches in the IPTC Caption and the special IPTC field "fsi_search_data". These are the places FSI Pages Converter adds the search keywords to. In order to search all text meta data fields available you can specify the following template instead: %%searchtext%%

Please refer to the FSI Server documentation for additional search options.

Please refer to the chapter → *Searching in FSI Pages* for further details.

SearchResultServerTemplate	
Description	Template for the search result
Syntax	String
Default	pages_searchresult" (FSI Server)
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	5.5.5 or higher

Specifies the server template to request in search queries.

SearchUseMethodGet	
Description	Use HTTP GET method to request search results
Syntax	Boolean
Default	false
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	4.0.0 or higher

By default FSI Pages requests search results using the HTTP POST method. Using this parameter you can force FSI Pages to use the HTTP GET method when requesting search results. Please refer to the chapter → *Searching in FSI Pages* for details.

SearchSortResults	
Description	Sort search results
Syntax	Boolean
Default	True
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	5.0.4

If set to true, the search results are ordered by page number.

If set to false, the search results are listed in the order the imaging server sends them to FSI Pages.

Events	
Description	Enable plug-in events
Syntax	Boolean
Default	false
Context	Pages plug-in node or HTTP query prefixed with "pages_"

Using this parameter you can enable or disable plug-in event notifications e.g. for the JS Bridge plug-in. Please see → *JavaScript Bridge Plug-in* for details.

FlipCornerSize	
Description	Size of interactive page corners
Syntax	Number
Default	10%
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	3.5.5 or higher

Specifies the size of the interactive page corners you can use to flip pages. You can either specify the size in pixels or in percent of the page width. Setting this parameter to "0" disables interactive page flipping by dragging the page corners.

FlipEdgeSize	
Description	Size of interactive page edges
Syntax	Number
Default	5%
Context	Pages plug-in node or HTTP query prefixed with "pages_"
Version	3.5.5 or higher

Specifies the size of the interactive outer page edges you can use to flip pages. You can either specify the size in pixels or in percent of the page width. Setting this parameter to "0" disables interactive page flipping by dragging the page edges.

10.1 FSI Pages mobile



The FSI Pages mobile application brings the core functionality of FSI Pages to Apple iOS and Google Android based mobile devices which do not support the Adobe Flash™ Plug-in. In contrast to FSI Pages which requires the Adobe Flash® Player, FSI Pages mobile is based on JavaScript and does not require the Adobe Flash® Player. Images will be displayed according to the way the mobile device is held (horizontal/vertical) using the single source imaging capabilities of FSI Server. FSI Pages mobile offers touch screen support.

How to embed FSI Pages mobile

The standard request to load FSI Pages is as follows:

```
http://fsi.domain.com/fsi/viewer/fsi.swf?cfg=pages_presets/catalog_links&[more_parameters]
```

The related request to load FSI Pages mobile would be:

```
http://fsi.domain.com/fsi/viewer/applications/pagesmobile/?cfg=pages_presets/catalog_links&[more_parameters]
```

Please note the "fsi.swf" is replaced by the path of your FSI Pages mobile installation, which is "applications/pagesmobile/" in the FSI Viewer folder.

Changing images on the image server will be visible for the end user immediately, as images are loaded on request and are not embedded into the application in contrast to other "apps" for mobile devices. FSI Pages Mobile uses the same configuration files and parameters as FSI Pages, but due to technical limitations not all parameters are supported. Unsupported parameters will be ignored by FSI Pages mobile, thus it does not require additional configuration to publish your images on mobile devices as well. Please refer to the FSI Pages parameter list to see whether a certain parameter is supported by FSI Pages mobile.

Requirements

Webkit based browser (Apple Safari on iOS or Google Chrome on Android devices)
FSI Server 2.0 or above

10.1.1 Pages Mobile Parameters

MobileDoublePageRatioLimit	
Description	Defines Page display behaviour
Syntax	String (Float or n%)
Default	0.7
Context	Pages
Version	FSI Pages mobile 1.0.0

If displaying two pages will automatically result in less than 70% (default) of the screen height being used, a single page is displayed instead of two pages for better space utilization. Setting this parameter to 100% will force single pages only, accordingly 0% will force double pages only.

InstructionsImageLandscape	
Description	Help image to be displayed left of the first page
Syntax	String
Default	_service/FSI Pages mobile/instructions/landscape.tif
Context	Pages
Version	FSI Pages mobile 1.0.0

An absolute or relative ImageURL that FSI Pages mobile displays left of the first page of the catalog in landscape format.

InstructionsImagePortrait	
Description	Help image to be displayed left of the first page
Syntax	String
Default	_service/FSI Pages mobile/instructions/portrait.tif
Context	Pages
Version	FSI Pages mobile 1.0.0

An absolute or relative ImageURL that FSI Pages mobile displays left of the first page of the catalog in portrait format.

MobileQuitButton	
Description	adds the option "Close Window" the the menu
Syntax	Boolean
Default	ture if window.opener exists
Context	FSI Pages mobile
Version	FSI Pages mobile 1.0.0

Adds the option "Close Window" to the menu.

DeferLinksToOpener	
Description	redirects links in FSI Pages mobile to the opener window
Syntax	Boolean
Default	false
Context	FSI Pages mobile
Version	FSI Pages mobile 1.0.0

Redirects links in FSI Pages mobile to the opener window.

AutoElevateToHTTPS	
Description	switches protocol to https
Syntax	Boolean
Default	true
Context	Pages
Version	FSI Pages mobile 1.1.17

Automatically switches the protocol used by the different Viewers to https if required by the setup.

10.1.2 Pages Mobile Plug-in Attribute targetdevice

```
<plugin src="imprint" targetdevice="desktop,mobile" />
```

Defines whether a plug-in should be loaded on the desktop Flash application, the mobile device, or both of them. Plug-ins have default values according to the availability on each platform, so this should not be changed unless you know what you are doing.

11 FSI Showcase Add-on

FSI Showcase Add-on integrates a resizable scroll pane to FSI Viewer displaying thumbnails of multiple images. Clicking a thumbnail displays the image in FSI Viewer.

This way a single viewer instance can be used to display hundreds of images presenting thumbnails and optional texts for each image.



11.1 Using FSI Showcase

The integration of FSI Showcase is very similar to an ordinary FSI plug-in. To enable the showcase you have to add a plug-in node to your (default) XML configuration file:

Integrating FSI Showcase

```
<Plugins>
...
    <Plugin src="showcase" basecolor="58A8FF"/>
...
</Plugins>
```

11.2 Defining Image Lists

The images to be displayed in FSI Showcase have to be defined in an `<images>` node added to your XML configuration file. The format and options of the collections is the same as the collections for FSI Pages Add-on.

There are three different ways of defining image lists:

1. Creating image nodes referencing external XML configuration files

External XML configuration files

```

<Images>
...
    <Image label="My First Image" file="image1.xml" />
    <Image label="My Second Image" file="image2.xml" />
...
</Images>

```

2. Creating image nodes containing complete image configuration**Complete XML configurations**

```

<Images>
...
    <Image label="My First Image">
        <FPX>
            <SRC value="image1.fpx" />
        </FPX>
    </Image>

    <Image label="My Second Image">
        <FPX>
            <SRC value="image2.fpx" />
            <Width value="8096" />
            <Height value="12300" />
        </FPX>
        <Options>
            <NoNav value="true" />
        </Options>
    </Image>
...
</Images>

```

3. Retrieving automatically generated image lists from a server**Complete XML configurations**

```

<Plugins>
...
    <Plugin src="showcase" server="..." dir="..." />
...
</Plugins>

```

Method 1 and Method 2 can be freely combined while Method 3 retrieves a complete image list ignoring previously defined <images> nodes. Please note that the format of image lists is the same as for FSI Pages. The chapter → *Parameters to retrieve Image Collections* on FSI Pages contains more detailed information on creating and retrieving.

Image Order

The order of the images within the thumbnail bar corresponds to the order of the <image> declarations within the <images> section.

Graphical Bookmarks and Product Tours

The Showcase does not re-load images if the image source did not change (e.g. "FPXSrc" and "Effects"). You can therefore use the Showcase to create graphical bookmarks or product tours by specifying the same image source ("FPXSrc") for multiple images and defining different "InitialView" parameters. The Showcase parameter → *UseInitialView* should be set to "true" (default) in this case.

FSI Parameters and FSI Showcase

All FSI parameters and FSI Plug-in parameters can be used with FSI Showcase Add-on. The only exception is the plug-in integration which will be described later on. The Showcase Add-on follows the hierarchy displayed below to build the configuration for an image:

1. Hierarchy of parameter definitions
2. Default Configuration File
3. Showcase Configuration File
4. Image Configuration File
5. Image Node
6. HTTP Query

This way you can define global parameters for all images by defining the parameters in

- 1 (default configuration)
- 2 (showcase configuration) or
- 5 by HTTP query.

Parameters specific to certain images have to be defined in the external image configuration file (3) or in the <image> node of the image list (4).

Due to the hierarchy displayed above parameters declared by query override all other parameter definitions. This can be especially useful if you want to override




parameters defined in external configuration files, for example overriding the "MenuAlign" parameter.

11.3 FSI Showcase Layout





The appearance of the Showcase Add-on can be customized by a large number of configuration parameters. The parameters will be described in detail in the sections "Basic Parameters" and "Advanced Parameters".

This section describes general layout considerations and measures.

Parameter Layout:

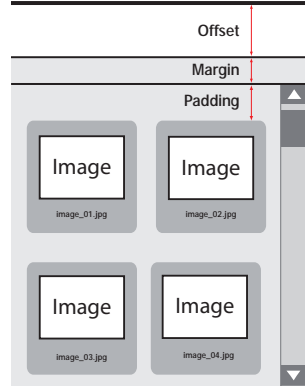
Outside	Inside	Floating
		
<i>Shrinks the size of the viewer by the initial size of the thumbnail bar</i>	<i>Shrinks the size of the image (view port) by the initial size of the thumbnail bar</i>	<i>Floats above the viewer without modifying the viewer or image size</i>

Parameter Align:

Right	Left	Top	Bottom
			

The figures to the left show the alignment with "Layout" set to "Outside" and "MenuAlign" set to "BL" (Bottom Left). You can achieve various layouts by modifying the 2 Showcase parameters ("Layout" and "Align") and the FSI Viewer parameter "MenuAlign".

In addition to the layout parameters explained above you can define Offsets, Margins and Paddings for each side of the Showcase. If you use the "Inside" or "Floating" layout you will most likely have to define an offset to avoid parts of the Showcase being hidden below the menu bar. The actual values depend on the FSI Skin being used. The figure to the right illustrates the parameters "OffsetTop", "MarginTop" and "PaddingTop". The values for the right, left and bottom side can be set accordingly.

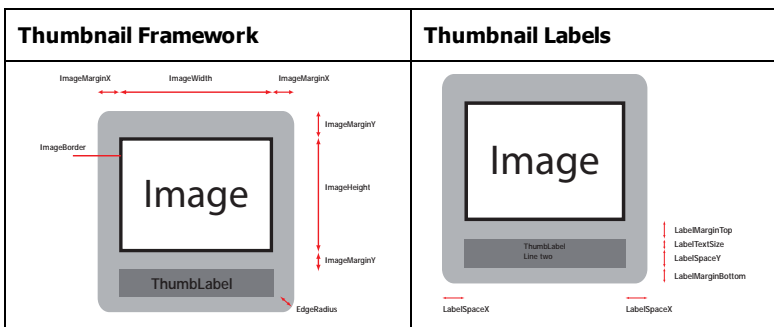


- Use Offset to offset the entire thumbnail bar
- Use Margin to offset the scroll bar and thumbnails
- Use Padding to offset the thumbnail



Using Offset values you can define small thumbnail bars that do not occupy the entire width or height of the viewer.

The chapter → *Advanced Parameters* provides a description of parameters which can be used to resize and/or colorize all elements of the thumbnail framework. If you just want to change the general color scheme you can use the parameter "BaseColor" instead. You can rescale the thumbnail using the parameter "BaseScale".



11.4 Showcase Parameters

The Showcase parameters described in the following can be provided in 3 different ways:

a) Attributes of the plug-in tag

Example – Parameter provided as attribute

```
<Plugins>
...
    <Plugin src="showcase" Layout="Outside" />
...
</Plugins>
```

b) Child nodes of the plug-in node

Example – Parameter provided as child nodes

```
<Plugins>
...
    <Plugin src="showcase" >
        <Layout value="Outside" />
        <Align value="Left" />
    </Plugin>
...
</Plugins>
```

c) By HTTP query value prefixed with "showcase_"

Example – Parameter provided by query

```
<Param name="movie" value="fsi.swf?showcase_layout=Outside">
```

The following values can be used:

Type	Example
Number	"90"
Pixel	"90"
String	"ZoomIn"
URL	"http://www.neptunelabs.com/"
Bool	either "0" / "1" or "true" / "false"
HexColor	"FF00FF"
Color Transition	"FF00FF, FFFFFFFF"

11.5 Basic Parameters

BaseColor	
Description	Base color of the thumbnail bar
Syntax	HexColor
Default	"DDDDDD" (light gray)
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Base color for the thumbnail frame work, the scroll bar and the background of the thumbnail bar. All color values will automatically be derived from this color if not defined separately (see → *Advanced Parameters*).

BaseScale	
Description	Scale of thumbnail images
Syntax	Number
Default	100
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Scale of thumbnails in percent, e.g. "150" for 150% of the original size. The user can additionally scale the thumbnails by clicking the corresponding buttons in the drag

bar menu. Please note that this value does not change the dimension of thumbnails being retrieved from the server (see ImageWidth and ImageHeight).

Layout	
Description	Defines the way FSI Showcase integrates into the viewer
Syntax	"outside", "inside" or "floating"
Default	"outside"
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

- **Outside (default)**
The showcase occupies the entire height or width of the viewer instance shrinking the viewer size by the initial size of the thumbnail bar. The thumbnail bar will be placed above all viewer elements when enlarging the thumb bar by default.
- **Inside**
The showcase reduces the view port of the viewer by the initial size of the thumbnail bar, leaving the menu elements unchanged. The thumbnail bar will be placed below all viewer elements when enlarging the thumb bar by default.
- **Floating**
The showcase floats above the view port without modifying the viewer or the view port size. The thumbnail bar will be placed below all viewer elements when enlarging the thumb bar by default.

Align	
Description	Position of the thumbnail bar
Syntax	"left", "top", "right", "bottom"
Default	"right"
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines the position of the thumbnail bar. Please note that the orientation of a possible scrollbar can be defined by the parameter "HScroll".

InitialSize	
Description	Initial size of the thumbnail bar
Syntax	Size in pixel or percent or in number of items
Default	"1 item"
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines the initial size of the thumbnail bar. You can define the size in absolute pixel values (e.g. "120"), in percent of the viewer size ("50%") or by the number of thumbnails being displayed horizontally (Align="left"/"right") or vertically (Align="top"/"bottom").

If you set the "Layout" parameter to "Outside" or "Inside" this value additionally defines the minimum size of the thumbnail bar the user can achieve by dragging the thumbnail bar.

HScroll	
Description	Display a horizontal scroll bar if required
Syntax	Boolean
Default	false
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

- false
Arrange thumbnails vertically and use a vertical scroll bar if required
- true
Arrange thumbnails horizontally and use a horizontal scroll bar if required

DragBar	
Description	Enable/Disable dynamic resizing of the thumbnail bar
Syntax	"Resizable", "Fixed" or "Hidden"
Default	"Resizable"
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

- Resizable
The user is able to resize the thumbnail bar dynamically. The width of the splitter bar of the showcase is increased.
- Fixed
The user is not able to resize the thumbnail bar dynamically. The width of the splitter bar of the showcase is normal.
- Hidden
The user is not able to resize the thumbnail bar dynamically and the splitter bar of the showcase is invisible.

DragBarWidth	
Description	Size of the drag bar
Syntax	Number
Default	8
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

A floating point value defining the width of the drag bar of the thumbnail bar.

DragMenu	
Description	Show/Hide the buttons in the thumbnail bar
Syntax	Boolean
Default	true
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Show the buttons of the thumbnail bar enabling the user to resize the thumbnails and the bar itself. This parameter is always "false" if the parameter "DragBar" is "hidden".

DragMenuWidth	
Description	Size of the menu of the drag bar
Syntax	Number
Default	9
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

A floating point value defining the size of the menu inside the drag bar.

UseInitialView	
Description	Use InitialView parameters for thumbnail images
Syntax	boolean
Default	true
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Use the image section defined by the parameter "InitialView" to create the thumbnail image if set to "true". If set to "False" the entire image will be used.

InitialImage	
Description	Image selected at startup
Syntax	Number
Default	1
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

The index of the image selected on startup. Enter the index of the image from 1 to n images.

ToolTips	
Description	Show tool tips when hovering above a thumbnail
Syntax	Boolean
Default	true
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Show tool tips when the mouse cursor hovers above a thumbnail. The text presented in the tool tip is the text of the thumbnail.

This option can be useful if you choose to hide the text on the thumbnail or if the thumbnail label is too long to be entirely displayed on the thumbnail itself. Multi-line texts are being presented separated by commas.

MinimizeOnSelect	
Description	Minimize the thumbnail bar on select
Syntax	Boolean
Default	true
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Minimize the thumbnail bar to the initial size when the user selects (clicks) a thumbnail.

TextOutsideFrame	
Description	Display text outside thumbnail frame
Syntax	Boolean
Default	false
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Display the thumbnail label inside the thumbnail frame work (false) or outside the frame work (true). The option only applies to thumbnails in normal state, it does not have an impact on active and selected thumbnails.

ThumbEffects	
Description	Image manipulation parameters applied to thumbnail images
Syntax	String
Default	—
Context	<Options>
Version	3.1.1 or higher

Applies image manipulation effects like "sharpen" or "quality" to all thumbnail images. Please refer to the → *Applying Effects to Specific Images Only* parameter for a detailed description.

ShowControls	
Description	Show or hide image changing buttons
Default	true
Context	<Options>
Version	3.1.1 or higher

If set to false, the buttons to switch to the next or previous pages are hidden.

11.6 ImageList Parameters

You can additionally use automatically generated showcase configuration files for a specific directory on FSI server. The generated showcase configuration contains all required data listing images in alphabetical order.

Experienced users might additionally edit the server template files used to generate the image list. The corresponding file is located in the 'WEB-INF/templates/list/' directory of your FSI Server:

Filename	Description
image_list.xml	Creates the <Images> section of the Showcase configuration

You can specify custom templates to create the image list using the "ListTemplate" parameter.

The following parameters have to be passed to the showcase in order to use automatic showcase configurations:

Dir	
Description	Image server Base directory
Syntax	String
Default	—
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Specifies the image directory on the imaging Server the <Images> section will be created for, e.g. 'images/paintings/'

Server	
Description	Path to imaging server
Syntax	String
Default	—
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Specifies the URL of the imaging server to generate the configuration. E.g. 'http://mydomain.com/fsi/'

This parameter can be omitted if the Showcase has been installed to the default location on the imaging server.

Please note: Loading configurations across domain boundaries will fail due to a security restriction of the Adobe Flash™ plug-in.

ListTemplate	
Description	Template to retrieve image lists
Syntax	String
Default	"fsi/image_list.xml"
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"
Version	3.2.6

Specifies the XML template to be used to retrieve the image list from imaging server. You can use custom list templates to retrieve image configurations files from imaging server, e.g. to including HotSpot definitions contained in IPTC data of the images.

EmptyImages	
Description	Define blank images (place holders)
Syntax	String
Default	---
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"
Version	3.3.0 or higher

Using this parameter you can add place holders to your image collection. The value of the parameter must contain one or more image indices the place holders shall occupy. E.g. `<EmptyImages value="2,10" />` adds place holders at position 2 and 10.

Alternatively you can add `<image empty="true" />` nodes to your image collection.

11.7 Advanced Parameters

The following parameters can be used to customize the appearance of the thumbnails and the thumbnail bar. Please note that you can easily change the color scheme by defining a BaseColor rather than defining all colors individually. To change the overall scale of thumbnails you can use the basic parameter "BaseScale".

Background	
Description	Color of the background of the thumbnail bar
Syntax	HexColor
Default	Defined by parameter BaseColor
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Color of the background of the thumbnail bar.

BackgroundAlpha	
Description	Opacity of the background of the thumbnail bar
Syntax	Number
Default	100
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines the opacity of the thumbnail bar background from "0" (invisible) to "100" (opaque).

ScrollBaseColor	
Description	Base color of the scroll bar
Syntax	HexColor
Default	Derived from BaseColor
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines a color for the scroll bar.

ScrollTrack	
Description	Color of the track bar of the scroll bar
Syntax	HexColor
Default	Derived from ScrollBaseColor
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines a color for the track area of the scroll bar.

ScrollArrow	
Description	Color of the arrows of the scroll bar
Syntax	HexColor
Default	Derived from ScrollBaseColor
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines a color for the arrows on the buttons of the scroll bar. Please note that the same color applies to the labels of the drag bar menu.

DragBarHighlight	
Description	Highlight color of the drag bar
Syntax	HexColor
Default	Derived from BaseColor
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"
Version	3.5.0 or higher

Defines the highlight color of the drag bar.

DragBarShadow	
Description	Shadow color of the drag bar
Syntax	HexColor
Default	Derived from BaseColor
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"
Version	3.5.0 or higher

Defines the shadow color of the drag bar.

DragBarRoundBevel	
Description	Use a round or small bevel for the drag bar
Syntax	Boolean
Default	true
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"
Version	3.5.0 or higher

Use a large, round bevel or a small, rectangular bevel when drawing the drag bar.

ImageHeight and ImageWidth	
Description	Dimension of thumbnail images
Syntax	Pixel
Default	45
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines the maximum physical dimension of thumbnail images in pixel. The actual size of thumbnail images depends on the aspect ratio of an image.

ThumbspacingX and ThumbspacingY	
Description	Space between thumbnails
Syntax	Pixel
Default	4
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

The spacing between the thumbnails in the thumbnail bar in horizontal and vertical direction.

ImageMarginX and ImageMarginY	
Description	Space around thumbnail image
Syntax	Pixel
Default	0
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Space between image border and thumbnail border left/right of the image (ImageMarginX) and above/below the image (ImageMarginY). If you choose round edges (parameter "EdgeRadius") this value will automatically be increased if required to ensure that the image does not overlap the thumbnail edge.

ImageBorderWidth	
Description	Thickness of the border around the thumbnail image
Syntax	Pixel
Default	1
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines the thickness of the border around the thumbnail image. Setting this parameter to "0" disables the image border.

ImageBorderHighlight	
Description	Highlight color of the border around the thumbnail image
Syntax	HexColor
Default	Derived from BaseColor
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines the color of the highlighted section of the image border (left and top).

ImageBorderShadow	
Description	Shadow color of the border around the thumbnail image
Syntax	HexColor
Default	Derived from BaseColor
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines the color of the darkened section of the image border (right and bottom).

ThumbFace	
Description	Color of the thumbnail face
Syntax	HexColor or color transition
Default	Color transition derived from BaseColor
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines the color of the thumb face.

This value can be either defined as a solid color (e.g. "00FF00") or as a linear color transition from top/left to bottom/right by entering two color values separated by comma (e.g. "FF0000,00FF00").

The corresponding parameters for active and selected state are: *ThumbActiveFace* and *ThumbSelectedFace*.

ThumbBorderWidth	
Description	Thickness of the thumbnail border
Syntax	Pixel
Default	1
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Thickness of the border around the thumbnail. Enter "0" to disable the border.

The corresponding parameters for active and selected state are:
ThumbBorderActiveWidth and *ThumbBorderSelectedWidth*.

ThumbBorderHighlight	
Description	Highlight color of the thumbnail border
Syntax	HexColor
Default	Color derived from BaseColor
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines the color of the highlighted section of the thumbnail border (left and top).
The corresponding parameters for active and selected state are:
ThumbBorderActiveHighlight and *ThumbBorderSelectedHighlight*.

ThumbBorderShadow	
Description	Shadow color of the thumbnail border
Syntax	HexColor
Default	Color derived from BaseColor
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines the color of the darkened section of the thumbnail border (right and bottom).

The corresponding parameters for active and selected state are:
ThumbBorderActiveShadow and *ThumbBorderSelectedShadow*.

EdgeRadius	
Description	Radius of rounded thumbnail edges
Syntax	Pixel
Default	6
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines a radius for drawing rounded edges of the thumbnail frame. Use "0" for straight edges and positive values for rounded edges.

OffsetBottom, OffsetLeft, OffsetRight, OffsetTop,	
Description	Offset from the top of the viewer instance
Syntax	Pixel
Default	0
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

This value defines the offset from the top of the viewer instance to the top of the thumbnail bar of the showcase. Please see the "Showcase measurements" figures for details.

MarginBottom, MarginLeft, MarginRight, MarginTop	
Description	Margin from the top of the thumbnail bar
Syntax	Pixel
Default	0
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

This value defines the margin in pixels from the top of the thumbnail bar to the content and scrollbar of the thumbnail bar of the showcase. Please see the "Showcase measurements" figures for details.

PaddingBottom, PaddingLeft, PaddingRight, PaddingTop	
Description	Padding of thumbnails inside the thumbnail bar
Syntax	Pixel
Default	0
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

This value defines the top padding of thumbnails inside the thumbnail bar of the showcase.

Please see the "Showcase measurements" figures for details.

HoverZoom	
Description	Zoom thumbnails "onMouseOver"
Syntax	Number
Default	100 (disabled)
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Zoom thumbnails when the user points at a thumbnail. The default value is "100" (no effect). Possible values are 100% (no effect) to 250%.

HoverRotation	
Description	Rotate thumbnails "onMouseOver"
Syntax	Number
Default	0 (disabled)
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Rotate thumbnails when the user points at a thumbnail. The default value is "0" (no effect). Possible values are 0 (degree) to 360 (degree).

The following parameters can be used to customize the text displayed on the thumbnails:

LabelTextSize	
Description	Text size of the thumbnail labels
Syntax	Pixel
Default	0 (no text)
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Text size in pixel for the text presented below the thumbnail image. Use "0" to disable text and positive values (e.g. "8") to enable texts on the thumbnails.

LabelMode	
Description	Content of the thumbnail labels
Syntax	Number
Default	0
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Obsolete. Please use LabelContent and LabelTemplates parameters described below instead.

LabelContent	
Description	Content of thumbnail labels
Syntax	String
Default	—
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

The specified string serves as a label for the thumbnail images in the thumbnail bar. The string may contain basic HTML tags (see → *HTML Tags available in FSI Viewer*) to format the text. Additionally you can add templates being replaced at runtime. Please see → *LabelTemplates* parameter below for details.

LabelTemplates	
Description	Runtime Templates for thumbnail labels
Syntax	String
Default	—
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

FSI Showcase add-on replaces the specified templates in the thumbnail labels (parameter "LabelContent") with the value specified in the image's configuration at runtime. This is: any configuration value (specified in _default.xml, image.xml or by HTTP Query) can be integrated into the thumbnail label. To do so, you need to:

- a) Add a corresponding template to the "LabelContent" parameter value
- b) Add the template to the "LabelTemplates" value

Each template needs to be entered as an XML node, while the node name is the name of the full configuration parameter name like you would use the parameter in the HTML query for FSI Viewer, e.g. <FPXWidth/> or <Showcase_Dir/>.

Thumbnail labels with runtime templates

```
<Plugin src="showcase">
...
<LabelContent><b><ImageIndex/>. <ImageLabel /></b><br/>Size:
<FPXWidth/> x <FPXHeight/> px</LabelContent>
<LabelTemplates value="FPXWidth,FPXHeight" />
...
</Plugin>
```

The resulting label with this configuration looks like this:

1. SomeImageName.tif
Size: 2048 x 1024 px

The two templates <ImageIndex/> and <ImageLabel /> are built-in templates containing the numeric index of an image and the "label" attribute of the <image> node of the image list.

LabelPrefix	
Description	Prefix applied to thumbnail texts
Syntax	String
Default	—
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

A prefix string applied to all thumbnail labels or "false" for no prefix.

LabelTextHeight and LabelTextWidth	
Description	Horizontal or vertical space for thumbnail labels
Syntax	Pixel
Default	—
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

You can use this value to increase the horizontal or vertical space available for thumbnail labels. In case the thumbnail label text does not fit into the available space, the label will be displayed cropped, but the entire label is available in the thumbnail's tool tip.

LabelTextAlign	
Description	Text alignment for all thumbnail labels
Syntax	left, center or right
Default	left
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Align the thumbnail labels left, centered, or right in the available space for label texts.

This value has no effect if the contents of a label contains `<p align="[alignment]"></p>` tags.

LabelTextFont	
Description	Font used for thumbnail label texts
Syntax	String
Default	"_sans"
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines the font used to display thumbnail labels. This value has no effect if the contents of a label contains `` tags. Please note that the font will only be reflected, if the font is available on the user's system. Otherwise a similar font will be displayed.

LabelTextColor	
Description	Text color of thumbnail labels
Syntax	HexColor
Default	000000
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

A 6-digit hexadecimal color value defining the text color of thumbnail labels. This value has no effect if the contents of a label contains `` tags.

LabelMarginBottom and LabelMarginTop	
Description	Margin above/below the thumbnail text
Syntax	Pixel
Default	0
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

You can use this value to add additional space above (LabelMarginTop) or below (LabelMarginBottom) the thumbnail text.

LabelOffsetTop	
Description	Adjust vertical position of the thumbnail text
Syntax	Pixel
Default	0
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

You can use this value to adjust the vertical position of the text area of the thumbnail. Negative values move the text area up while positive values move the text area down.

LabelOnRight	
Description	Display thumbnail labels right of the image
Syntax	Boolean
Default	false
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"
Version	4.1.3

Set this parameter to "true" to display thumbnail labels right of the images instead of below the thumbnail images.

LabelOffsetLeft	
Description	Adjust horizontal position of thumbnail text
Syntax	Pixel
Default	10
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"
Version	4.1.3

You can use this value to adjust the horizontal position of the text area of the thumbnail. Negative values move the text area to the left while positive values move the text area to the right.

PreloadThumbs	
Description	Load all thumbnail images on startup
Syntax	Number
Default	0
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"

Setting the value to "0" makes the showcase load thumbnails on-demand. Setting positive values enables the pre-loader which loads all images on startup. The given value specifies the number of images to preload simultaneously.

Please note: Be careful when activating this option especially with automatically generated image lists or large image lists as it might cause higher server loads and increased (and possibly unnecessary) network traffic.

CropThumbnails	
Description	Crop images to match thumbnail dimension
Syntax	Boolean
Default	false
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"
Version	3.5.0 or higher

Automatically crop images to match the dimensions of the thumbnails.

KeepImageSection	
Description	Preserve the view when selecting thumbnails
Syntax	Boolean
Default	false
Context	Showcase plug-in node or HTTP query prefixed with "showcase_"
Version	3.5.0 or higher

Usually FSI Showcase displays the entire image (or the image section specified by the parameter "InitialView") when selecting a thumbnail. Set this parameter to "true" to if you want to preserve the image section you are currently viewing when selecting another image.

12 Plug-in Reference

This part of the documentation lists all standard FSI Viewer plug-ins. The purpose of FSI Viewer plug-ins is integrating functionality at runtime that might not be required in general. This way users need to download only that part of FSI Viewer code necessary to provide the required functionality. Additionally the plug-in system provides the possibility to add functionality required by an individual customer or by a certain group of customers. Please contact your FSI Viewer dealer in case you are missing any kind of functionality.

Plug-in Extends

The Plug-in Extends section lists the available plug-ins for FSI Viewer and Addons and what each plug-in extends. Please note that any plug-in that extends FSI Viewer may also affect FSI Pages in page zoom mode, as page zoom mode is FSI Viewer technically.

12.1 AddThis

Plug-in Extends:

- FSI Pages
- FSI Pages mobile

Plug-in Location:

/plugins/addthis.swf

Function:

Social Platform Sharing Tool

Syntax:

```
<plugin src="addthis"/>
```



Plug-in Parameters

UserName	
Description	Username to login the AddThis service
Syntax	String
Default	—
Context	Plug-in tag

This value defines your AddThis username to login the AddThis service.

TargetFrame	
Description	defines the HTML frame to open the AddThis page in
Syntax	String
Default	—
Context	Plug-in tag

This value defines the HTML frame to open the AddThis page in.

DialogWidth	
Description	defines the width of the dialog window
Syntax	Number
Default	480
Context	Plug-in tag

Defines the width of the dialog window.

DialogHeight	
Description	defines the height of the dialog window
Syntax	Number
Default	430
Context	Plug-in tag

Defines the height of the dialog window.

ShowLabels	
Description	defines whether the service labels should be displayed
Syntax	Boolean
Default	false
Context	Plug-in tag

This value defines whether the service names in the dialog should be displayed.

ShowTooltips	
Description	defines whether tooltips should be displayed.
Syntax	Boolean
Default	false
Context	Plug-in tag

This value defines whether tooltips describing the services should be displayed.

"true" if showLabels == false and "false" if showlabels == true

BookmarkTitle	
Description	defines the title for the bookmark
Syntax	String
Default	"auto:document"
Context	Plug-in tag

This value defines the title for the bookmark (default: "auto:document" which is the title of the container HTML document)

BookmarkURL	
Description	defines the URL to bookmark
Syntax	String
Default	"auto:document"
Context	Plug-in tag

This value defines the URL to bookmark (default: "auto:document" which is the URL of the container HTML document). For further information please see the section "Details on the BookmarkURL" below.

Services	
Description	defines which services should be used
Syntax	String
Default	all
Context	Plug-in tag

Comma separated string: defines which services should be used.
To get a list of all service codes available, please see the "listServices" parameter or visit the AddThis website.

ListServices	
Description	lists all available services
Syntax	Boolean
Default	false
Context	Plug-in tag

List all available services and priorityServices in plug-ins section of the debug window.

PriorityServices	
Description	defines which services to display first
Syntax	String
Default	see below
Context	Plug-in tag

Comma separated string: services to display first
 Default: "facebook_like,google_plusone_share,twitter,email,stumbleupon,reddit,tumblr,pinterest,linkedin,blogger,delicious,wordpress,friendfeed,livejournal"

RemoveServices	
Description	defines whether tooltips should be displayed
Syntax	String
Default	-
Context	Plug-in tag

Comma separated string: services to hide in the list
 Example: "twitter,hyves,google,email,facebook,linkedin"

AppendFSIParameters	
Description	automatically add FSI Viewer parameters to the BookmarkURL
Syntax	Boolean
Default	true
Context	Plug-in tag

Automatically add FSI Viewer parameters to the BookmarkURL (e.g. pages_InitialPage with the value of the current page)

ShareURLTemplate	
Description	template to create the AddThis URL to call
Syntax	String
Default	see below
Context	Plug-in tag

Template to create the AddThis URL to call.

Default:

`http://api.addthis.com/oexchange/0.8/forward/ [SERVICEID] /offer?url= [URL] &title=[TITLE]&username=[USERNAME]`

Library	
Description	defines the SWF file containing service icons and details
Syntax	XML
Default	addthis_lib.swf
Context	Plug-in tag

This value defines the SWF file containing service icons and details.

Details on the BoomarkURL:

By default the plug-in uses the URL of the container HTML document if possible. You can on the other hand specify any other URL.

The "AppendFSIParameters" (default:true) specifies whether to add FSI Viewer parameters dynamically to the BookmarkURL. The plug-in will then add/modify the query parameter "FSIPassthrough" to the URL. The value of this query parameter contains the automatically added FSI Viewer parameters you should pass on to the FSI Viewer instance, e.g. using Javascript or a server side scripting language.

This way you can not only bookmark an HTML document containing a specific FSI Viewer instance, but you can instead bookmark a specific page of the FSI Pages instance.

All you need to do is extracting the "FSIPassthrough" query value from the documents URL and add this parameters to the FSI Viewer instance (e.g. in the writeFlashCode() Javascript call).

12.2 BackgroundImage



Plug-in Extends:

- FSI Viewer
- FSI Showcase
- FSI Pages
- FSI Pages mobile

Plug-in Location:

/plugins/backgroundimage.swf

Function:

Adds a background image to FSI Viewer

Syntax:

```
<Plugin src="backgroundimage"
FPXSrc="images/bgimage.tif" />
```



This plug-in adds a background image to FSI Viewer. Usually background images are used with FSI Pages only, as with FSI Viewer and FSI Showcase most of the background image is covered by the zoomable image. The plug-in crops the specified image so that the entire area of FSI Viewer instance is filled with the background image.

Plug-in Parameters

FPXSrc	
Description	Relative path to the image on your imaging server
Syntax	String
Default	—
Context	Plug-in tag

The path to an image on your imaging server to be used as a background. If you use a relative path (recommended) the FSIBase parameter will be used to get the absolute path to the image.

Effects	
Description	Optional effect parameters to pass to the imaging server
Syntax	String
Default	—
Context	Plug-in tag

Like with the → *Effects* parameter for FSI Viewer itself you can optionally pass image effect parameters to your imaging server that affect the appearance of the background image.

12.3 Chapters



Plug-in Extends:

- FSI Pages
- FSI Pages mobile

Plug-in Location:

/plugins/chapters.swf

Function:

Adds a select box to access parts of a catalog directly.

Syntax:

```
<Plugin src="chapters"/>
```

CHAPTER

--- Select ---	▼
Woman.....	4
Man.....	25
Sports Wear.....	35
Shoes.....	47
Kids.....	52
Home.....	63

This plug-in adds a select box to the user interface of FSI Pages. The user can directly access sections of a catalog by selecting the corresponding item in the select box. You can assign colors and different levels to each chapter using simple XML based data describing the document structures.

Example: XML data for the Chapters plug-in

```
<Indexdata>
  <Index label="First Page" page="1" color="FFFF00" />
  <Index label="Chapter 1" page="10" color="00FF00">
    <Index label="Chapter 1.1"
      page="20" color="00FF00"/>
    <Index label="Chapter 1.2"
      page="30" color="00FF00"/>
    <Index label="Chapter 1.3"
      page="40" color="00FF00"/>
  </Index>
  <Index label="Chapter 2" page="50" color="0000FF"/>
  <Index label="Chapter 3" page="60" color="CCCCCC"/>
  <Index label="Chapter 4" page="70" color="FF0000"/>
</Indexdata>
```

Plug-in Parameters

IndexData	
Description	XML document structure data
Syntax	XML Data
Default	---
Context	Plug-in attribute

XML Data describing the document structure. Add the XML data as a child node of the "Chapters" plug-in node. Alternatively you can specify the XML data using an external XML file.

IndexDataFile	
Description	URL to an external XML file describing the document structure
Syntax	String
Default	---
Context	Plug-in attribute

Alternatively to the "IndexData" parameter you can use an external XML file to provide the XML document structure data required for this plug-in. The XML file must be located in the same domain as FSI Viewer to avoid cross-domain security restrictions. If you enter a relative path, FSI Viewer loads the file from the subdirectory "pages_url" of your FSI Viewer configuration directory ("FSIBase") If you store XML files in the directory "[fsi]/config/pages_xml/" you therefore only need to specify the file name of the XML file to load.

Please note: You need to use an external XML file in UTF-8 format when using special language specific characters (characters \geq 0x80).

With version 2016 of FSI Server, it is possible to set this parameter to "auto" in order to request the information from the server instead of loading the information from the file.

PageNumbers	
Description	Display a starting and ending pages of chapters
Syntax	Boolean
Default	true
Context	Plug-in attribute

Add the starting and ending page of a chapter to the chapter label.

ComboLabelBackgroundAlpha	
Description	Background opacity of the select box background
Syntax	Number
Default	10
Context	Plug-in attribute

Opacity of the select box background from 0 (transparent) to 100 (opaque).

ColoredIndex	
Description	Display colored index markers in the select box
Syntax	Boolean
Default	true
Context	Plug-in attribute

Display the color specified in the document structure XML data in front of the chapter labels.

IndexMarginWidth	
Description	Width of the colored page index
Syntax	Number
Default	0
Context	Plug-in attribute

Floating point value specifying the width of the colored index located on the outer page margin in percent of the page width. Enter 0 to disable the colored page margin.

Please note: This value requires setting "ColoredIndex" to true as well.

Width	
Description	Width of the select box
Syntax	Pixels
Default	200 (depending on skin)
Context	Plug-in attribute

Usually the FSI Pages skin defines the width and position of the select box. Alternatively you can specify a fixed width in pixels using this parameter.

Sort	
Description	Sort index items by page number
Syntax	Boolean
Default	true
Context	Plug-in attribute
Version	4.1.5 or above

By default the plug-in sorts the index items ascending by page number. Set this parameter to "false" to use the order in which the index items are listed in the XML file.

Update	
Description	Update the selected index on page flip
Syntax	Boolean
Default	true
Context	Plug-in attribute
Version	4.1.5 or above

By default the plug-in automatically selects the matching index item after a page flip. Set this parameter to "false" to display the first index item always, e.g. to make the plug-in display "Go to section ..." always.

ValidateData	
Description	Use valid index items only
Syntax	Boolean
Default	true
Context	Plug-in attribute
Version	4.1.5 or above

By default the plug-in does not display items that refer to page numbers higher than the last page or lower than first page. Set this parameter to display all index items, even if refer to non-existing pages.

Index Item Attributes

Each <index> node represents an item in the chapters drop down referring the user to a page in your document. The following attributes need to or can be added to each <index> item:

Attribute	Description
page	Index of the page to refer to (required)
label	Title for a chapter (required)
file	File to download for a chapter (from version 4.0.0) See chapter → <i>Saving Pages</i> for details.
color	6-digit hexcolor for use with "ColoredIndex" parameter
musicSong	An absolute or relative URL to an MP3 file to play for this chapter (from version 4.0.7). Please note that you need to add the plug-in "music" as well in this case.

12.4 ClockProgress



Plug-in Extends:

- FSI Viewer
- FSI Showcase

Plug-in Location:

/plugins/clockprogress.swf

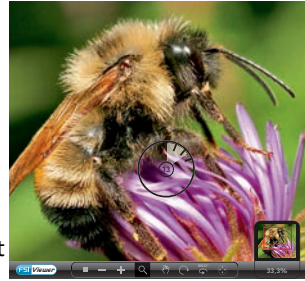
Function:

Replacement for default progress bar

Syntax:

```
<Plugin src="clockprogress" />
```

The "ClockProgress" plug-in replaces the default progress bar with a round progress indicator.



Plug-in Parameters

The clockprogress plug-in can be customized by adding attributes to the <plugin> tag.

E.g. <Plugin src="clockprogress" alpha="20"/>

The following attributes can be used to customize the clockprogress plug-in:

Alpha	
Description	Opacity of progress indicator
Syntax	Number
Default	50
Context	Plug-in attributes

Defines the opacity of the progress indicator. Possible values range from 0 to 100.

0: transparent
100: opaque

Size	
Description	Size of progress indicator
Syntax	Number
Default	60
Context	Plug-in Tag

Defines the overall size of the progress indicator in pixel.

Color1	
Description	Color of the area representing the amount of pending data
Syntax	HexColor
Default	000000
Context	Plug-in attributes

A 6-digit hexadecimal number defining the color of the area representing the amount of pending data.

Color2	
Description	Color of the area representing the amount of data already loaded
Syntax	HexColor
Default	FFFFFF
Context	Plug-in attributes

A 6-digit hexadecimal number defining the color of the area representing the amount of data already loaded.

LineColor	
Description	Line color
Syntax	HexColor
Default	000000
Context	Plug-in attribute

A 6-digit hexadecimal number defining the color of the lines.

TextColor	
Description	Text color
Syntax	HexColor
Default	000000
Context	Plug-in attribute

A 6-digit hexadecimal number defining the color of the text.

posX	
Description	Horizontal Position
Syntax	String / Number
Default	Depends on MenuAlign parameter
Context	Plug-in attribute

Defines the horizontal position of the progress indicator.

Possible values:

C	Centered horizontally
Number > 0	Offset from left
Number < 0	Offset from right

posY	
Description	Vertical position
Syntax	String / Number
Default	Depends on MenuAlign parameter
Context	Plug-in attribute

Defines the vertical position of the progress indicator.

Possible values:

C	Centered vertically
Number > 0	Offset from top
Number < 0	Offset from bottom

12.5 ColorAdjust



Plug-in Extends:

- FSI Viewer
- FSI Showcase
- FSI Pages

Plug-in Location:

/plugins/coloradjust.swf

Function:

Adjusts the color level of images

Syntax:

```
<Plugin src="coloradjust" />
```



Plug-in Parameters

AutoHideDialog	
Description	Auto Hide Dialog
Syntax	Boolean
Default	true
Context	ColorAdjust
Version	5.0.0

Hides the interface dialog when changing values.

Blue, Green, Red	
Description	Blue value, Green value, Red value
Syntax	Float
Default	100
Context	ColorAdjust
Version	5.0.0

Amount of blue, green and red.

Brightness	
Description	Brightness of image
Syntax	Float
Default	100
Context	ColorAdjust
Version	5.0.0

The image brightness.

MenuOffset	
Description	MenuOffset
Syntax	Number (Pixels)
Default	—
Context	ColorAdjust
Version	5.0.0

Defines the menu offset.

SetValuesForEachImage	
Description	Save Values for each image seperately
Syntax	Boolean
Default	false
Context	ColorAdjust
Version	5.0.0

If enabled, the color values can be set for each image. If disabled (default) the values are used for all images.

ShowUI	
Description	Show Interface
Syntax	Boolean
Default	true
Context	ColorAdjust
Version	5.0.0

If disabled, the user interface will be hidden, enabling the effects.

12.6 ContextMenu



Plug-in Extends:

- FSI Viewer
- FSI Showcase
- FSI Pages

Plug-in Location:

/plugins/contextmenu.swf

Function:

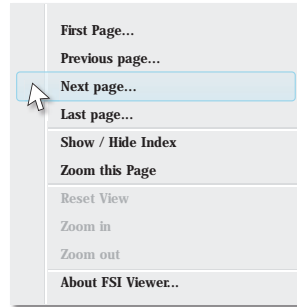
Adds FSI Viewer related items to the Flash player's context menu

Requirements:

Flash player 7 or above

Syntax:

```
<Plugin src="contextmenu" />
```



This plug-in does not provide any parameters. You can on the other hand translate the menu items or additional items. The example below shows the built-in configuration of this plug-in.

Sample configuration

```

<Plugin src="contextmenu" >
  <MenuItems>
    <!-- FSI PAGES ONLY -->
    <Item label="First Page..." action="FirstPage" />
    <Item label="Previous page..." action="PreviousPage" />
    <Item label="Next page..." action="NextPage" />
    <Item label="Last Page..." action="LastPage" />
    <Separator />
    <Item label="Show / Hide Index"
    action="ShowHidePageIndex"/>
    <Item label="Zoom this Page" action="ZoomPage"/>
    <Separator />
    <!-- FSI SHOWCASE ONLY -->
    <Item label="Previous Image" action="PreviousImage"/>
    <Item label="Next Image" action="NextImage"/>
    <Separator />
    <!-- FSI VIEWER -->
    <Item label="Reset View" action="Reset"/>
    <Item label="Zoom in" action="ZoomIn"/>
    <Item label="Zoom out" action="ZoomOut"/>
  </MenuItems>
</Plugin>

```

Plug-in Parameters

MenuItems	
Description	Refers to the XML describing the context menu entries
Syntax	String
Default	—
Context	Plug-in tag
Version	5.0.0

Defines the XML snippet describing the context menu.

12.7 CustomButton



Plug-in Extends:

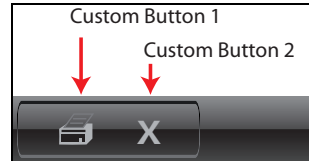
- FSI Viewer
- FSI Showcase

Plug-in Location:

/plugins/custombutton.swf

Function:

Adds custom buttons to the FSI Viewer menu bar



Syntax:

```
<Plugin src="CustomButton" buttons="btn1,btn2" />
```

The "CustomButton" plug-in can be used to integrate one or more custom buttons into the menu bar of FSI viewer. Clicking a button invokes the URL specified in the plug-in parameter. This way you can integrate buttons performing a JavaScript call or opening up a web page.

Plug-in Parameters

MenuOffset	
Description	Offset in pixel to previous menu elements
Syntax	Number in pixel
Default	0
Context	Plug-in tag

Defines an offset of all custom buttons in pixels to elements of the menu bar.

Buttons	
Description	Comma separated list of unique identifiers
Syntax	String
Default	—
Context	Plug-in tag

Enter a list of unique identifiers separated by commas. Each identifier represents a new button that can be configured using the parameters below. E.g. "btn1,btn2" adds two custom buttons. To specify parameters for these buttons you have to prefix the parameters by the unique identifier plus, e.g. "btn1.Url" defines the URL to use for the button named "btn1".

[ID].Offset	
Description	Offset of the button in pixel to previous (custom) buttons
Syntax	Number in pixels
Default	0
Context	Plug-in tag

Specifies an offset for the button named "[ID]" in pixel to previous (custom) buttons.

[ID].URL	
Description	Url to open on click
Syntax	String
Default	—
Context	Plug-in tag

Specifies an URL to open when the user clicks the button named "[ID]". You can specify a URL to a web page or alternatively a script call, e.g. "javascript: void alert ("foo");" Please note: You can use templates to be replaced by FSI Viewer at runtime. The button appears disabled if no URL is specified.

[ID].Frame	
Description	HTML frame to open the specified URL in
Syntax	String
Default	_self
Context	Plug-in tag






Specifies the HTML frame in which to open the specified URL for the button named "[ID]". Besides using an identifier of a frame inside your HTML frame set you can as well use one of the predefined values "_self", "_top", "_blank" or "_parent".

[ID].ToolTip	
Description	Tool tip text
Syntax	String
Default	—
Context	Plug-in tag

Specifies the tool tip to be displayed when the user places the cursor above the custom button named "[ID]".

[ID].LabelFrame	
Description	Symbol displayed on the button
Syntax	Number
Default	1
Context	Plug-in Tag

Specifies the symbol to be displayed on top of the custom button named "[ID]". The predefined symbols are:

GFX	ID	Example
	1	Information
	2	Close
	3	Resize or Restore
	4	Open Window
	5	Print

You can add your own graphics to be used as a label for custom buttons. You will need Adobe Flash™ editor to add custom icons.

Using Templates in URL parameters

Starting from FSI version 3.2.3 you can use templates when defining URLs for custom buttons. These templates will be replaced by FSI Viewer when the user clicks a custom button. This way you can retrieve parameters from FSI Viewer dynamically and pass the values to a JavaScript function or a server side script.

Templates in CustomButton URLs

```
<Plugin src="custombutton" buttons="btn1"
btn1.URL="javascript:alert ('$$cfg.FPXWidth$$ x
$$cfg.FPXHeight$$')" />
```

The example above outputs the size of the source image via JavaScript.

The list below shows available templates for custom button URLs:

Template in URL	Replaced by...
\$\$cfg. [Name]\$\$	The value of the FSI Viewer parameter [name]. Be sure to use the long parameter names only (e.g. "FPXWidth" instead of "Width"). You can retrieve parameter values retrieved from the imaging server using this template as well. E.g. "\$\$cfg.FPXWidth\$\$" returns the size of the source image – whether specified manually or retrieved from the imaging server.
\$\$View\$\$	A string representing the current image section in FSI Viewer (see Parameter → <i>InitialView</i> for details)
\$\$URLView\$\$	The complete URL to the image currently displayed in FSI Viewer. The resulting string contains the templates "\$\$width\$\$", "\$\$height\$\$" and "\$\$template\$\$" which need to be replaced in JavaScript or server side script.
\$\$URLScene\$\$	Same as above, but representing the entire scene image.
\$\$DownloadRatio\$\$	Aspect ratio representing the factor "width/height". You can use this value to calculate the width of the image to download.
\$\$Zoom\$\$	Current magnification relative to the initial size of the image.
\$\$Selection\$\$	A string representing the current selection (requires plug-in SelectFrame)
\$\$Measure\$\$	A string representing the current measurement (requires plug-in Measure)

12.8 Fullscreen



Plug-in Extends:

- FSI Viewer
- FSI Showcase
- FSI Pages

- FSI Viewer JS
- FSI Showcase JS

Plug-in Location for Flash viewers:

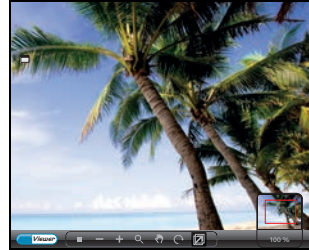
/plugins/fullscreen.swf

Function:

switch to fullscreen mode

Syntax:

```
<Plugin src="fullscreen" />
```



If using the object tag when embedding FSI Viewer, please note the parameter allowfullscreen has to be set as shown.

HTML integration without considering old browsers

```
<Object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/
swflash.cab#version=6,0,65,0" width="[Width]" height="[Height]"
">
    <PARAM name="movie" VALUE="[URL and Parameter]">
    <PARAM name="bgcolor" VALUE="[Background color]">
    <PARAM name="menu" VALUE="false">
    <PARAM name="allowfullsreen" VALUE="true" />
</Object>
```



Please note that the Adobe Flash™ Player prohibits keyboard input in full screen mode.

Integration for JS viewers:

Adding the parameter `plugins:"FullScreen, Resize"` enables a full screen view option in FSI Viewer JS. The `Resize` parameter needs to be set in order to ensure proper adjusting to the full screen.

NOTE: If the FSI Viewer JS is embedded in an `iFrame`, the attribute `allowfullscreen="true"` needs to be added to the `iFrame`. Otherwise the FullScreen Plug-in will not work due to security restrictions of the browser.

12.9 History



Plug-in Extends:

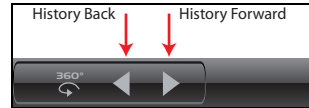
- FSI Viewer
- FSI Showcase
- FSI Pages

Plug-in Location:

/plugins/history.swf

Function:

History functionality to jump to previously viewed image sections



Syntax:

```
<Plugin src="history" length="25" />
```

The "History" plug-in extends the user interface by 2 buttons, "History Back" and "History Forward". Using these two buttons the user is able to navigate through image sections previously viewed.

Plug-in Parameters

MenuOffset	
Description	Indentation of the menu buttons
Syntax	Number
Default	0
Context	Plug-in Tag

Specifies the space in pixels left of the menu button(s) of the plug-in.

Length	
Description	Length of the history list
Syntax	Number
Default	10
Context	Plug-in tag

The number of image sections that will be stored in the view history from 1 to 99.

If the list exceeds the amount of entries defined by the "length" parameter, the first entry of the list will be replaced.

Loop	
Description	Loop the history list
Syntax	Boolean
Default	false
Context	Plug-in tag

If this value is set to "true" the viewer will display the first viewed section when exceeding the last entry of the history and the last entry when stepping before the first entry of the history list.

Providing the default value "false", the history will not be looped, this is the history list stops at the first and the last position.

SmallButtons	
Description	Size of menu buttons
Syntax	Boolean
Default	true
Context	Plug-in tag

If this value is set to "false" the buttons of this plug-in will be displayed in the default button size. Otherwise the buttons appear slightly smaller than other buttons.

12.10 HotSpots



Plug-in Extends:

- FSI Viewer
- FSI Showcase

- FSI Viewer JS

Plug-in Location:

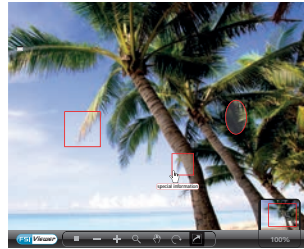
/plugins/hotspots.swf

Function:

By defining clickable areas (hotspots) the user is able to interact with the image by clicking on areas inside the zoomable image.

Syntax:

```
<Plugin src="hotspots" />
```



The "HotSpots" plug-in adds an image map like functionality to the FSI Viewer. The plug-in adds a button to the menu bar which allows the user to show or hide the HotSpots. The clickable areas are defined via XML inside the image specific *.xml file and enable an increased interaction with the Viewer.

There are 4 different combinable actions that can be assigned to each hotspot:

- Display a tool tip
- Zoom to the area defined by the hotspot
- Open a HTML page
- Restart the Viewer with a different image / configuration

Adding Hotspot Functionality to your Images

Integrating the Plug-in

To integrate the HotSpot plug-in into the viewer you have to add the following line to the <Plugins> section of either the _default.fsi file or to an image specific .fsi configuration file:

```
<Plugin src="hotspots" />
```

Defining the HotSpots Section

To define HotSpots for an image you have to add an XML-section to your image specific .fsi file. The definition syntax is similar to image maps for static images inside a HTML page.

Each HotSpot is being defined by a single XML node inside the <HotSpots> group.

The basic structure of a HotSpot definition is the following:

```

HotSpot definition
<Plugin src="hotspots" visible="true">
    <HotSpots>
        <circle spot="" tip="" url="" />
        <rect spot="" tip="" url="" />
    </HotSpots>
</Plugin>

```

Alternatively to adding the hotspot XML section as a child node of the HotSpot <plugin> node, you can add the XML section as a root child in the same level as <FPX> or <Options> in a configuration file.

NOTE: For adding hotspots to FSI Viewer JS, please look into the → <hotspots3d> Section.

Plug-in Parameters

Visible	
Description	Initial visibility of the HotSpots
Syntax	Bool
Default	false
Context	Plug-in attribute or <HotSpots> group tag

Defines the initial visibility of HotSpots.

The HotSpot button state will be set accordingly.

NOTE: also applies to FSI Viewer JS. All other parameters below solely apply to the Flash based viewers. For HotSpot configuration in FSI Viewer JS, please go to: → <hotspots3d> Section

Alpha	
Description	Defines the opacity for all HotSpots
Syntax	Number between 0-100
Default	100
Context	Plug-in attribute or <HotSpots> tag

Defines the opacity for all HotSpots from 0 (invisible) to 100 (opaque).

BaseURL	
Description	Prefix for relative HotSpot URLs
Syntax	String
Default	—
Context	Plug-in attribute or <HotSpots> group tag

Defines a prefix that will be applied to all HotSpots containing relative URLs. This way you can omit e.g. the domain in URL-attributes of subsequent HotSpots.

Events	
Description	Enable JSBridge events for HotSpots
Syntax	Boolean
Default	false
Context	HotSpots
Version	5.0.4

Enable to fire event notifications for HotSpots via JSBridge. The following events will be fired:

- 1) hotspots_MouseOver
- 2) hotspots_MouseOut
- 3) hotspots_MousePress
- 4) hotspots_MouseRelease

The event value for event 1-3 is the hot spot's tool tip or – if defined – the hot spot node's "id" attribute.

The event value for event 4 depends on the hot spot action and can be:

- the value of the "URL" attribute
- the area to zoom to
- the "cfg" parameter of a new configuration file to load
- the value of the "SelectImage" attribute

Defining Default Attributes

The following attributes define default attributes to all HotSpot that do not contain a corresponding attribute itself.

E.g. defining a DefaultTarget of "_self" will open all hyperlinks in the frame the viewer resides in, provided that there is no individual Target attribute defined in a hotspot node.

DefaultTarget	
Description	Default target for hyperlinks
Syntax	String
Default	_blank
Context	Plug-in attribute or <HotSpots> group tag

Defines the default target(frame) for HotSpot hyperlinks.

Valid parameters are "_blank", "_self", "_top", "_parent" and names of frames inside a HTML frameset.

DefaultMove	
Description	Zoom to HotSpot area by default
Syntax	Bool
Default	true
Context	Plug-in attribute or <HotSpots> group tag

Defines the default value for all subsequent HotSpots that do not contain the "Move" attribute.

DefaultSkew	
Description	Permit skewing of HotSpots
Syntax	Bool
Default	false
Context	Plug-in attribute or <HotSpots> tag

Defines the default value for all subsequent HotSpots that do not contain the "Skew" attribute.

DefaultColor	
Description	Default color of HotSpot borders
Syntax	HexColor
Default	FF0000
Context	Plug-in attribute or <HotSpots> tag

Defines the default color of HotSpot borders.

DefaultFill	
Description	Fill the HotSpot area by default
Syntax	Boolean
Default	true
Context	Plug-in attribute or <HotSpots> group tag

Defines the default value for all subsequent HotSpots that do not contain the "Fill" attribute.

If this value is set to false, all HotSpots will be shown as outlines by default.

DefaultFillColor	
Description	Default fill color for the HotSpot areas
Syntax	HexColor
Default	FFFFFF (white)
Context	Plug-in attribute or <HotSpots> group tag

Defines the default fill color for HotSpots that do not contain the "FillColor" attribute.

DefaultFillAlpha	
Description	Default fill opacity for the HotSpot areas
Syntax	Number
Default	25
Context	Plug-in attribute or <HotSpots> group tag

Defines the default fill opacity for HotSpots that do not contain the "FillAlpha" attribute where "0" is transparent and "100" fully opaque.

Defining Individual HotSpots

Each HotSpot definition must at least contain the name of the HotSpot shape and the "spot" attribute which defines the position of the HotSpot. The tag name of each definition specifies the shape of the HotSpot.

Available default shapes include:

- rect (Square / Rectangle)
- circle (Circle / Ellipse)
- star (Star)
- polygon (Polygon of n straight lines)

HotSpot definition

```

<HotSpots>

<circle spot="1, 1, 0.35842, 0.15383, 0.61261, 0.40802, 0"
tip="circle />
  <rect spot="1, 10, 0.54314, 0.68446, 0.74036,
0.88168, -56"
wtip="rectangle" url="rect.htm" />
  <polygon coords="0.5,0 ,1,0 ,1,0.5, 0.5,0.5" >
  <text>some <b>formatted</b> text</text>
</polygon>

</HotSpots>

```

Please refer to the chapter → *Hotspot Authoring* for useful tips on how to create HotSpots.

HotSpot Parameters

The following attributes can be assigned to each HotSpot:

Spot	
Description	Position of the HotSpot
Syntax	String
Default	—
Context	HotSpot tag

Defines the position of the HotSpot.

It is recommended to use the plug-in → *SelectFrame* to obtain this value. Please refer to the chapter → *Hotspot Authoring* for more information on this parameter.

Coords	
Description	Coordinates for polygon shapes
Syntax	String
Default	—
Context	HotSpot tag

Defines the coordinates of a polygon shape. In contrast to predefined shapes, polygons do not require the "spot" parameter. Instead you need to provide the x and y coordinates (0..1) for each point determining the polygon shape. You might want to add the spot parameter to specify the scene and sceneset the polygon applies to with 3D presentations.

View	
Description	Image section to display
Syntax	String
Default	Value of spot parameter
Context	HotSpot tag
Version	3.2.3 and above

Defines the position to display when move is set to true. If this parameter is undefined, the value of spot will be used instead. Please refer to the chapter → *Hotspot Authoring* for more information on this parameter.

Tip	
Description	Tooltip
Syntax	String
Default	—
Context	HotSpot tag

Defines a string to be used for a tooltip when the user moves the cursor over the HotSpot.

For long texts or "HTML formatted" texts (see → *HTML Tags available in FSI Viewer*) you can add a <text> child node to each hotspot node instead of using the "Tip" parameter.

HotSpot text definition
<pre><HotSpots> ... <Circle spot="1,1,0,0,0.6,0.4,0" > <text>Headline
Some text</text> </Circle> ... </HotSpots></pre>

Move	
Description	Zoom to HotSpot area on click
Syntax	Boolean
Default	false
Context	HotSpot tag

Zoom to the image section covered by the HotSpot, if the user clicks the HotSpot. You can use the "DefaultMove" parameter to change the default behavior.

Skew	
Description	Allow skewing of the HotSpot area
Syntax	Boolean
Default	false
Context	HotSpot tag

If this parameter is set to false the width of the hotspot area will equal the height of the HotSpot regardless of the values provided with the "Spot" parameter. You might want to use this parameter to ensure that e.g. HotSpots defined with the <circle> tag look like circles instead of ellipses. You can use the "DefaultSkew" parameter to change the default behavior.

Color	
Description	Color of the HotSpot border
Syntax	HexColor
Default	FF0000
Context	HotSpot tag

Defines the color of the HotSpot border.

The value has to be a 6 digit hexadecimal number like "FFFF00" for yellow or "0000FF" for blue. You can use the "DefaultColor" parameter to change the default border color.

Fill	
Description	Fill the HotSpot area
Syntax	Bool
Default	false
Context	HotSpot tag

If this value is set to false, the HotSpot will be shown as an outline.

You can use the "DefaultFill" parameter to change the default behavior.

URL	
Description	Hyperlink of the HotSpot
Syntax	URL
Default	—
Context	HotSpot tag

Defines a hyperlink to be opened if the user clicks the HotSpot.

The URL can be given relatively or absolutely.

The Target parameter (or respectively the "DefaultTarget" parameter) defines the targetframe for the hyperlink. Please refer to the "BaseURL" parameter as well.

Target	
Description	Target frame for hyperlinks
Syntax	String
Default	_blank
Context	HotSpot tag

Defines the targetframe for HotSpot-hyperlinks.

Valid parameters include "_blank", "_self", "_top", "_parent" and names of frames inside a HTML frameset.

Please refer to the "DefaultTarget" parameter for details on how to set a default target for hyperlinks.

CFG	
Description	FSI parameters to be used on click
Syntax	String (query)
Default	—
Context	HotSpot tag

If you define a CFG parameter for a HotSpot the viewer will be reinitialized with the given configuration if the user clicks the HotSpot.

This can be useful to switch from one image to another if the user clicks a HotSpot.

The syntax of the CFG parameter has to consist of FSI Parameters concatenated with the ampersand character (like a HTTP query).

```
Parameter1=value1&Parameter2=value2...
```

Example:

```
cfg="cfg=image2&NoNav=1"
```

Defining the parameter like this, the viewer would reinitialize using the configuration file "image2.xml" providing the FSI parameter "NoNav" with a value of "true".

Please refer to the FSI Viewer manual for a list of all FSI Parameters and for details on how to create a query string that provides FSI parameters.

SelectImage	
Description	Select another image (FSI Showcase only)
Syntax	Number
Default	—
Context	HotSpot tag

Clicking the HotSpot selects another image inside FSI Showcase. The index can be specified absolutely by providing a number only. The first image is being represented by "0".

Alternatively you can select an image relatively from the currently selected image. To select an image relatively you have to provide a number prefixed by "+" or "-", e.g. "+1" selects the next image.

Hotspot Authoring

The following section provides tips on how to obtain the string value required to define the "Spot" and the optional "View" parameter of a HotSpot.

The Spot parameter consists of 7 numbers concatenated with commas and provides the scene set, the scene, the image section (as a rectangle) and the rotation.

Example: 1, 10, 0.19056, 0.33579, 0.44862, 0.59385, 35

scene set 1, scene 10, rectangle [0.19056, 0.33579, 0.44862, 0.59385], rotation 35 °

Although you can define hotspot parameters manually, it is recommended to use the FSI Server web interface to edit hotspots.

Obtaining Hot Spot positions

Include the plug-in "SelectFrame" by adding the following parameters to the "_default.xml" or any other configuration file.

Including the SelectFrame plug-in

```
<Plugins>

  <Plugin src="SelectFrame" visible="1" window="1"
    rotate="1" AspectRatio="1:1" />

</Plugins>
```

After including the plug-in you will be able to author hotspots directly in FSI Viewer.

- You can use the value displayed in the "Selection" window for the "Spot" parameter of the hotspot.
- You can use the value displayed in the "View window for the optional "View" parameter.

When modifying the select box please use:

SHIFT	to lock the aspect ratio of the selection
CTRL	to resize the selection relative to the center
ALT	to rotate the selection
ALT + SHIFT	to rotate the selection in steps of 5 degrees

Defining Polygon HotSpots

Besides using predefined shapes as hotspots you can additionally add polygons. In this case you need to specify the X and Y coordinates for each point of the polygon using the "coords" attribute of the hotspot node.

HotSpot Polygon definition

```
<HotSpots>

  <Polygon spot="1,1" coords="0.5,0 ,1,0 ,1,0.5, 0.5,0.5" />

</HotSpots>
```

Please note that the "spot" attribute is required only, if you want to specify the target scene and/or scene set for the hotspot when using a 3D image.

The "coords" attribute contains [0..1] floating point coordinates where 0 is left or top and 1 is right or bottom. The coordinates for the polygon points defining the polygon shape need to be specified in the form:

x1, y1, x2, y2, x3, y3 ...

Theoretically you can use any number of coordinates (points), but keep in mind that more coordinates and more polygons require more CPU power on the user's system. Using very complex polygons with hundreds of coordinates will lead to poor performance as the polygons will be drawn by FSI Viewer at runtime.

HotSpot Z-Order

When defining HotSpots that overlap each other, you have to keep the z-order of the HotSpots in mind. The z-order of HotSpots is determined by the sequence in which they are defined in the configuration file, from bottom to top.

As a rule of thumb you should define big Hot Spot areas first to ensure that smaller Hot Spots are not completely hidden beneath.

12.11 Imprint

Plug-in Extends:

- FSI Pages mobile



Plug-in Location:

/plugins/imprint.swf

Function:

Displays information

Syntax:

```
<plugin src="imprint"
datafile="imprint.html" />
```



Plug-in Parameters

DataFile	
Description	URL to an external XML file describing the library structure
Syntax	String
Default	—
Context	Plug-in tag

The HTML file "imprint.html" will be displayed when the user opens the imprint window in FSI Pages mobile.

Please note that the HTML file must be an XHTML fragment, which means there must not be any <head>...</head> or <body>..</body> tags defined.

Please see the FSI Pages mobile Technotes for styling instructions on www.neptunelabs.com.

Please note: You need to use an external XML file in UTF-8 format when using special language specific characters (characters >= 0x80).

HTML Sample

```

<content>

<style type="text/css">
#imprint .x-html, #imprint .x-html
*{
    font-size: small;
    color:#cccccc;
    vertical-align:top;
}
</style>

<h1 style="font-size:large">Imprint</h1><hr>
<!--
other HTML code
//-->
</content>

```

AddButton	
Description	defines whether the plug-in icon is displayed
Syntax	Boolean
Default	false if FSI Pages is activated, otherwise true
Context	Plug-in tag

Defines whether the plug-in icon is displayed in the FSI Viewer tool bar. Defaults to FALSE if FSI Pages is activated.

MenuOffset	
Description	defines offset of all custom buttons in pixels to elements of menu bar.
Syntax	Number
Default	0
Context	Plug-in tag

Defines an offset of all custom buttons in pixels to elements of the menu bar.

DialogWidth	
Description	defines the width of the dialog window
Syntax	Number
Default	480
Context	Plug-in tag

Defines the width of the dialog window.

DialogHeight	
Description	defines the height of the dialog window
Syntax	Number
Default	430
Context	Plug-in tag

Defines the height of the dialog window.

12.12 JavaScript Bridge



Plug-in Extends:

- FSI Viewer
- FSI Showcase
- FSI Pages

Plug-in Location:

/plugins/jsbridge.swf

Function:

Integrates a JavaScript interface to enable interaction between FSI Viewer and JavaScript contained in the hosting HTML page.



Syntax:

```
<Plugin src="jsbridge" />
```

Requirements:

Browser supporting live connect between Flash and JavaScript (Livewire)



Please note: FSI Pages does not send JavaScript Events by default. Please set the FSI Pages parameter "events" to activate JSBridge interactivity.

The "JavaScript Bridge" plug-in provides a scripting interface to control FSI Viewer using JavaScript. Using the JavaScript bridge you can control viewer functions like "ZoomIn" and "Reset", you can remotely press/release Buttons inside the FSI Viewer and you can zoom to specified image sections directly. Additionally you can implement a JavaScript function receiving FSI event notifications.

Though the integration of the plug-in is very easy, basic knowledge of JavaScript is required to use JSBridge plug-in – especially regarding browser dependent JavaScript implementations.

Implementing the JavaScript Bridge plug-in

1) Add an ID-attribute to your FSI Viewer Object and Embed tag

```
<object id="fsiviewer" classid= ...>
<embed NAME="fsiviewer"...>
```

This way you are able to access the viewer object via Javascript.

2) Provide the appropriate action via SetVariable command

To control FSI Viewer from JavaScript you have to pass the appropriate command by setting the "FSICMD" variable of the FSI Viewer movie clip using the SetVariable () function.

Example: Button "Zoom In" (MSIE)

```
<input type="button" value="Zoom In"
onMouseDown="document.all.fsiviewer.SetVariable('FSICMD',
'ZoomIn');"
onMouseUp  ="document.all.fsiviewer.SetVariable('FSICMD', '');
>
```

Command Parameters

There are 3 different command types you can pass to FSI Viewer, FSI Pages or FSI Showcase:

1. Initiate an action directly, e.g. "ZoomIn" or "Reset"
2. Press and release of FSI Viewer buttons, e.g. "Button:Mouse_Mode_0"
3. Zoom to a specified image section, e.g. "Goto:1,6,0.4,0.2,0.6,0.4"
Please see the → Hotspot Authoring section for details on how to obtain valid image section parameters.

The following table lists possible values you can pass to FSI Viewer.

Please note that especially regarding the "Button:" command there might be more or not all commands listed available, depending on your FSI Viewer skin and integrated plug-ins.

All command strings have to be provided exactly as stated in the lists below, especially regarding upper-/ lowercase writing and white spaces.

1) Direct Commands

Usage: `FSIViewer.SetVariable('FSICMD', '[command]');`

To stop the initiated action use: `FSIViewer.SetVariable('FSICMD', ',');`

Command	Description
" (empty string)	Stop current action / Release Button
Reset	Reset Viewer
ZoomIn	Start zooming in
ZoomOut	Start zooming out
Up	Start panning upwards
Down	Start panning downwards
Left	Start panning to the left
Right	Start panning to the right
RotateLeft	Start rotating counter-clockwise (Z-axis)
RotateRight	Start rotating clockwise (Z-axis)
NextScene	Start rotating clockwise (Y-axis)
PreviousScene	Start rotating counter-clockwise (Y-axis)
NextSceneSet	Start rotating clockwise (X-axis)
PreviousSceneSet	Start rotating counter-clockwise (X-axis)
__DebugWindow	Show / Hide the debug window (if available)
__InfoWindow	Show / Hide the "about"-window (if available)

FSI Showcase only	Description
PreviousImage	Select previous Image
NextImage	Select next Image
SelectImage	Select image n Important: Use SetVariable("newImageIndex", n) where n is [0 .. images] prior to sending this command Alternatively you can select an image relatively from the currently selected image. To select an image relatively you have to set "newImageIndex" to a number prefixed by "+" or "-", e.g. "+1" selects the next image.
SwitchThumbs	Show / Hide Thumb bar

FSI Pages only	Description
FirstPage	Go to first page.
LastPage	Go to last page.
NextPage	Go to next page.
PreviousPage	Go to previous page.
GotoPage	Go to page n Important: Use <code>SetVariable("newImageIndex", n)</code> where n is [0 .. pages] prior to sending this command Alternatively you can specify a page relative to the current page. To go to a page relatively you have to set "newImageIndex" to a number prefixed by "+" or "-", e.g. "+1" is the same as "NextPage".
Print	Open or close the print dialog.
Save	Open or close the save dialog.
Search	Open or close the search dialog.
ZoomLeftPage	Zoom the left page
ZoomRightPage	Zoom the right page
NormalPageView	Exit zoom view
ShowHidePageIndex	Show or hide the page index (thumbnail view)

2) Button press / release

Using this command remotely presses/releases a button in FSI Viewer / FSI Showcase. There can not be more than one button pressed at a time. Pressing another button (by script) will automatically release the previous button.

Usage:

```
FSIViewer.SetVariable('FSICMD', 'Button:[ButtonID]');
```

To release the button previously pressed use:

```
FSIViewer.SetVariable('FSICMD', '');
```

ButtonID	Description
" (empty string)	Release previously pressed button
Mouse_Mode_n	<p>Activate mouse mode n .</p> <p>0 – Zoom 1 – Pan 2 – Rotate 3D 3 – Rotate (Z-axis)</p> <p>There might be more mouse modes available depending on loaded plug-ins</p>
Tool_Reset	"Reset" button, see command "Reset"
Tool_ZoomIn	"Zoom In" button, see command "ZoomIn"
Tool_ZoomOut	"Zoom Out" button, see command "ZoomOut"
Tool_SwitchUI	Show / Hide FSI Viewer user interface
Plugin_SwitchMusic	Toggle music on or off
Plugin_ExtendedNavigation	Toggle display of ExtendedNavigation plug-in
Plugin_HistoryBack	Click "History Back" button of the history plug-in
Plugin_HistoryForward	Click "History Forward" button of the history plug-in

Identifying commands of buttons not listed in the table

There might be some button identifiers that are not listed below, especially buttons integrated by other plug-ins. To obtain command identifiers of these buttons follow the steps below:

1. Activate the debug window of the FSI viewer (set parameter "debug" to "true").
2. Set the explore parameter of the jsbridge plug-in to "true".
3. Click the desired button inside FSI Viewer and copy the identifier from the debug window

3) Navigating to image sections

To navigate to a specific image section you simply use the "Goto" command followed by a valid image section parameter. Please refer to the chapter → *Debug Mode* for details on how to obtain image section parameters.

Example: Button "Zoom to image section" (MSIE)

```
<input type="button" value="Zoom to section"
onClick="document.all.fsiviewer.SetVariable('FSICMD',
'Goto:1,3,0,0,0');">
```

Implementing JavaScript callbacks

Using JavaScript callbacks enables you to react to FSI Viewer, FSI Pages or FSI Showcase events by executing JavaScript code. The JSBridge plug-in calls a JavaScript function using the Flash FSCommand architecture providing an event identifier and a parameter.

Please follow the steps below to implement JavaScript callbacks from FSI Viewer:

Step 1. Enable JavaScript callbacks by setting the plug-in parameter "Callback" to "true", e.g.

Example: Enabling JavaScript Callbacks

```
<plugin src="jsbridge" callback="true" />

Or alternatively via HTTP query:

...fsi.swf?cfg=foo&jsbridge_callback=1
```

Step 2. Add a unique identifier to the <object> and the <embed> tag of FSI Viewer / FSI Showcase.

Example: Adding ID / NAME parameters

```
<object id="fsiviewer" classid= ...>
<Param name="movie" value="...>
...
<embed NAME="fsiviewer"...>
...
</object>
```


Step 3. Add a JavaScript function to your HTML page receiving the event notifications via FSCCommand like in the example below.

IMPORTANT

The function name has to be exactly like in the example below and it has to start with the ID/NAME parameter specified in step 2.

If you do not use the ID/NAME "fsiviewer" you have to replace the corresponding script sections in the example below (printed bold) by your ID/NAME parameter.

Example: JavaScript Callback Function

```
<script language="javascript" type="text/javascript">

function fsiviewer_DoFSCCommand(fsi_event, params) {
    switch(fsi_event){
        case "LoadProgress":
            ...
            break;
    }
}

// Hook for Internet Explorer
if (navigator.appName &&
navigator.appName.indexOf("Microsoft") != -1 &&
navigator.userAgent.indexOf("Windows") != -1 &&
navigator.userAgent.indexOf("Windows 3.1") == -1) {
    document.write('<SCRIPT LANGUAGE=VBScript> \n');
    document.write('on error resume next \n');
    document.write('Sub fsiviewer_FSCCommand
(ByVal fsi_event, ByVal params)\n');
    document.write('    call fsiviewer_DoFSCCommand(fsi_event,
params)\n');
    document.write('end sub\n');
    document.write('</SCRIPT> \n');
}
</script>
```

After following the 3 steps described before you will receive the following event notifications:

Event (fsi_event)	Parameter (params)	Description
InitComplete	—	The viewer has finished parsing the configuration and is about to start loading image data.
ImageInfo	String	Provides concatenated image information. *See details below this table.
LoadingComplete	—	The initial load progress of an image completed.
LoadProgress	Number (percent)	Number corresponding to the progress bar inside FSI Viewer.
Press	Button ID	The button "ID" has been pressed.
Release	Button ID	The button "ID" has been released.
ToolTip	String ("%<escaped")	The tool tip "String" is about to be displayed. *See details below this table.
View	Image Section	The image section currently displayed. Can be used with the "Goto" command.
ImageUrl	URL template	Provides a template that can be used to build an image URL corresponding to the image being displayed in FSI Viewer. *See details below this table.
Action	Action ID	The action "ID" is currently being executed.
Modal	true / false	The viewer enters a passive

Event (fsi_event)	Parameter (params)	Description
		(true) or an active (false) state.
MouseMode	Mouse Mode ID	The mouse mode "ID" has been activated.
Zoom	Magnification (percent)	The magnification has changed.
ImageSelected	Image index (0..nImages)	A new image has been selected. (FSI Showcase only)

FSI Pages only

Event (fsi_event)	Parameter (params)	Description
onPagesInitComplete	Integer	Called after the FSI Pages Add-on is ready for use. The parameter provides the total number of pages.
onPagesGotoPage	Integer	Called each time before the user goes to a different page. The parameter provides the target page number.
onPagesPageChanged	Integer	Called each time after the current page has changed. The parameter provides the index of the page currently visible.
onPagesProgress	Integer	Called each time the load progress of FSI Pages changes. The parameter provides the number of pages remaining in the load queue.
onPagesShowIndex	Boolean	Called each time the user shows (true) or hides (false) the page index.
onPagesShowZoom	Boolean	Called each time the user starts (true) or ends (false) page zoom.
onPagesLinkClick	URL	Called each time the user clicks a link. The parameter contains the URL.

Please note that some plug-ins provide additional events. Please refer to the "events" parameter of the corresponding plug-in for details.

The recommended way to react to an event is to extend the switch...case block of the `fsiviewer_DoFSCommand (fsi_event, params)` callback function introduced in Step 3.

Detailed Event Parameter Description

INFO

The "Info" event parameter provides attributes of the source image currently being displayed.

Example:

```
width=8000;height=6000;TilesX=4;TilesY=3;SceneSets=;ImageIndex=4
```

The attribute "ImageIndex" applies to FSI Showcase or FSI Pages only and provides the 0-based index of the image currently selected.

ImageURL

To receive this notification you have to set the "ImageURLs" parameter of the plug-in to "true".

(Please refer to "Plug-in Parameters" below for details)

The "ImageURL" event parameter provides an URL template that can be used to retrieve the image currently displayed in FSI Viewer.

Example:

```
http://imageserver.domain/fsi/server?source=images/zoomimage.f  
px &width=[width]&height=[height]&left=0&top=0&right=1&bottom=1  
&template=[template]
```

The URL template can for example be used to set the "src" attribute of an tag.

Prior to using the template you have to replace the place holders (bold, in brackets) with valid values.

The **[template]** place holder applies to FSI Servers only.

ToolTip

The "ToolTip" event parameter provides the tool tip text that is about to be displayed. The values depend on the tool tip language selected. You have to use the JavaScript function **decodeURIComponent ()** before displaying this parameter.

Plug-in Parameters

AllowDomains	
Description:	Sub(domains) allowed to access the JSBridge interface
Syntax:	Domain list separated by ","
Default:	Subdomain of FSI Viewer, "www.domain.tld", "domain.tld", "*.domain.tld"
Context:	Plug-in attribute
Version:	4.1.0 or above

With Adobe Flash version 8 an additional limitation regarding Javascript to Flash access has been introduced. Starting with version 8, only a script from the same subdomain as the Flash movie (e.g. "fsi.domain.com") is allowed to access Flash / the JSBridge interface.

With FSI Viewer and FSI server it is very likely that you use FSI Viewer from "fsi.yourdomain.tld" while your HTML document resides in the subdomain "www.yourdomain.tld". Therefore JSBridge plug-in allows script access from the subdomain "www.yourdomain.tld" and from the main domain "yourdomain.tld" by default.

In case you need to access JSBridge from other (sub)domains you can specify a list of subdomains separated by ", "using this parameter, e.g.: <allowdomains value="www.domain.com,www.domain2.com"/>

Explore	
Description:	Activate debug output of Button IDs
Syntax:	Boolean
Default:	false
Context:	Plug-in attribute

If enabled the debug window will display the Button ID of each button pressed inside FSI Viewer. This way you are able to determine Button IDs not listed in this documentation. Additionally the debug window will output each command received by the JSBridge plug-in.

Callback	
Description:	Enable JavaScript callbacks
Syntax:	Boolean
Default:	false
Context:	Plug-in attribute

If enabled the events of FSI Viewer, FSI Pages or FSI Showcase will be forwarded to a specific JavaScript function.

Please refer to the section → *Implementing JavaScript callbacks* for details.

ImageUrls	
Description:	Provide ImageUrl templates with JavaScript callbacks
Syntax:	Boolean
Default:	false
Context:	Plug-in attribute

This parameter depends on the "Callback" parameter above. If "Callback" and "ImageUrls" parameter have been set to "true", "ImageUrl" notifications providing image URL templates will be sent to the JavaScript callback function.

Please refer to the "ImageURL" callback identifier for details.

12.13 LargeToolTips



Plug-in Extends:

- FSI Pages
- FSI Pages mobile

Plug-in Location:

/plugins/largetooltips.swf

Function:

Display multi-line tool tips with HTML formatted text.

Syntax:

```
<Plugin src="largetooltips"/>
```

This plug-in adds multi-line tool tips displaying HTML formatted text to links in FSI Pages add-on. The following HTML tags can be used in the tool tip text data:

```
<b>...</b> <i>...</i> <u>...</u>
```

```
<br/>
```

```
<font size="n" color="#RRGGBB" >...</font>
```

```
<p align="...">...</p>
```

```

```



Please note:

Certain limitations apply when using the img-tag: The width and height attributes are mandatory and the URL in the src-attribute must end width .jpg or .png. If you are using a single source imaging server to provide these images, then consider appending something like &ext=.jpg to the src-URL. Furthermore images will always be inserted into the textflow, if it is desired to continue the text below the inserted image, extra line breaks will be needed.

Plug-in Parameters

Width	
Description	Width of the tool tip box
Syntax	Pixels
Default	200
Context	Plug-in node or HTTP query prefixed with "argettooltips_"

Width of the tool tip box. The height of the box will be adjusted according to the length of the contained text.

Delay	
Description	Delay before displaying a tool tip
Syntax	Number (milliseconds)
Default	600
Context	Plug-in node or HTTP query prefixed with "argettooltips_"

You can specify a delay in milliseconds (1000 milliseconds = 1 sec) after pointing at a link area and before displaying a tool tip. Use 0 to display the tool tips immediately without a delay.

Shadow	
Description	Display a drop shadow
Syntax	Boolean
Default	true
Context	Plug-in node or HTTP query prefixed with "argettooltips_"

Display (true) or hide (false) a drop shadow for the tool tip box.

BackgroundColor	
Description	Background color of the tool tip box
Syntax	HexColor
Default	FFFFFF
Context	Plug-in node or HTTP query prefixed with "argetooltips_"

Background color of the tool tip box.

BackgroundAlpha	
Description	Opacity of the background of the tool tip box
Syntax	Number
Default	90
Context	Plug-in node or HTTP query prefixed with "argetooltips_"

Defines the opacity of the tool tip background from 0 (invisible) to 100 (opaque).

HighlightColor	
Description	Border highlight color
Syntax	HexColor
Default	999999
Context	Plug-in node or HTTP query prefixed with "argetooltips_"

6-digit hexadecimal color value specifying the highlighted (top-left) part of the tip box border.

ShadowColor	
Description	Border shadow color
Syntax	HexColor
Default	999999
Context	Plug-in node or HTTP query prefixed with "argetooltips_"

6-digit hexadecimal color value specifying the shadow (bottom-right) part of the tip box border.

LineWidth	
Description	Thickness of the tip box border
Syntax	Number
Default	2
Context	Plug-in node or HTTP query prefixed with "argetooltips_"

Thickness of the tip box border in pixels.

TextFont	
Description	Defines the font of the tool tip text
Syntax	String
Context	Plug-in node or HTTP query prefixed with "argetooltips_"
Version	5.0.0

Defines the font of the tool tip text. Please note that the desired font must be available on the users machine.

You need to specify the font name (e.g. "Courier New" or use one of the Flash Player predefinitions "_sans", "_serif" or "_typewriter")

TextColor	
Description	Default text color
Syntax	HexColor
Default	000000
Context	Plug-in node or HTTP query prefixed with "argetooltips_"

Default text color of text in the tip box. To use different colors in a tool tip you can use the HTML tag in the tool tip text data.

TextSize	
Description	Default text size
Syntax	Number
Default	11
Context	Plug-in node or HTTP query prefixed with "argettooltips_"

Default text size of text in the tip box. To use different sizes in a tool tip you can use the HTML tag `` in the tool tip text data.

CornerRadius	
Description	Radius of box corners
Syntax	Pixel
Default	20
Context	Plug-in node or HTTP query prefixed with "argettooltips_"

Radius of the box corners. Use "0" for straight edges and positive values for rounded edges.

12.14 Magnifier



Plug-in Extends:

- FSI Viewer
- FSI Showcase

Plug-in Location:

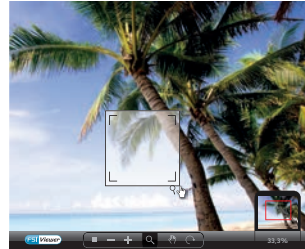
/plugins/magnifier.swf

Function:

Adds a magnifier to the user interface

Syntax:

```
<Plugin src="magnifier"
visible="true" size="10" />
```



The "Magnifier" plug-in adds a magnifier to the user interface that can be dragged using the mouse. Using the magnifier plug-in does not cause additional data traffic, as the plug-in enlarges already loaded image data by software.

The magnifier can be hidden and shown using a corresponding button in the menu bar. The level of magnification and the magnifier size can be defined by plug-in parameters.

Plug-in Parameters

MenuOffset	
Description	Indentation of the menu button
Syntax	Number
Default	0
Context	Plug-in attribute

Specifies the space in pixels left of the menu button of the plug-in.

Visible	
Description	Initial state of visibility
Syntax	Boolean
Default	false
Context	Plug-in attribute

Defines the initial visibility of the magnifier.

Size	
Description	Size of magnifier
Syntax	Number
Default	10
Context	Plug-in attribute

Defines the overall size of the magnifier. Possible values are between 4 and 16. Please note that greater values require more CPU power.

Magnification	
Description	Level of magnification
Syntax	Number
Default	2
Context	Plug-in attribute

Defines the level of magnification where 1 means no magnification, 2 means a magnification of 200% and so on. The maximum level of magnification is 8.

Reflections	
Description	Show or hide glass reflections
Syntax	Bool
Default	true
Context	Plug-in attribute

Show (true) or hide (false) glass reflections of the magnifier.

Handle	
Description	Show or hide magnifier handle
Syntax	Bool
Default	true
Context	Plug-in attribute

Show (true) or hide the handle of the magnifier glass.

HandleSize	
Description	Size of the handle in percent
Syntax	Number
Default	100
Context	Plug-in attribute

Size of the magnifier handle.

12.15 MaxZoom

1:1

Plug-in Extends:

- FSI Viewer
- FSI Showcase

Plug-in Location:

/plugins/maxzoom.swf

Function:

Zoom to physical resolution of the image.

Syntax:

```
<Plugin src="maxzoom" />
```

The "MaxZoom" plug-in expands the user interface by a button to quickly zoom to the physical resolution of the source image.



Plug-in Parameters

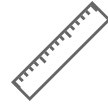
MenuOffset	
Description	Indentation of the menu button
Syntax	Number
Default	0
Context	Plug-in attribute

Specifies the space in pixels left of the menu button of the plug-in.

12.16 Measure

Plug-in Extends:

- FSI Viewer
- FSI Showcase



Plug-in Location:

/plugins/measure.swf

Function:

Provides distance and angle measuring



Syntax:

```
<Plugin src="measure" />
```

The Measure plug-in provides an additional mouse mode which enables the user to measure distances and angles inside the FSI Viewer by clicking & dragging.

Pressing SHIFT locks the angle to 45 degree steps.

Pressing CTRL moves the measurement line.

Mouse Mode

The mouse mode id for the measuring mode is 100.

You can use this id to set the InitialMouseMode parameter of the FSI Viewer.

e.g.

```
<InitialMouseMode value="100" />
```

or

```
?cfg=foo&InitialMouseMode=100
```

Setting up the Plug-in

The Measure Plug-in requires the real width of the entire image to enable distance measuring.

Example 1:

If you already know the entire width of the image simply add the "ImageWidth" and the "Suffix" parameter to the plug-in tag:

If the width of the entire image is 120.5 inches the corresponding plug-in tag looks as follows:

```
<plugin src="measure" ImageWidth="120.5" Suffix=" inches" />
```

Please note:

You might as well define all plug-in parameters via query or inside the <options> group by using the prefix "measure_".

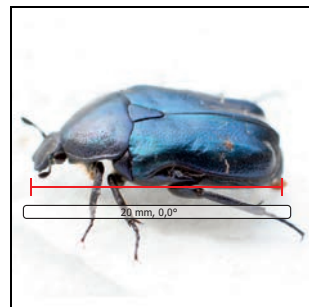
E.g.:

```
?cfg=image&Measure_ImageWidth=120.5&Measure_Suffix=%20inches
```

Example 2:

If you know the width of a part of the image, but you do not know the width of the entire image, you should follow the steps below:

1. Add the measure plug-in without any parameters.
<Plugin src="measure" />
2. Open the image inside the FSI Viewer and choose the "Measure" mouse mode.
3. Measure the part of the image you know the real width of.
(The Measure plug-in will display a decimal number n between 0 and 1)
4. The value w for the ImageWidth parameter is:
 $w = \text{RealWidth} / n$.
The Measure plug-in displays $n=20$.
The real length L of the bug is 53 mm.
ImageWidth $w = L / n = 50 / 20 = 2,65$



The corresponding plug-in tag is the following:

```
<plugin src="measure" ImageWidth="2.65" Suffix=" mm" />
```

Please note:

You might as well define all plug-in parameters via query or inside the <options> group by using the prefix "measure_".

E.g.:

```
?cfg=engine&Measure_ImageWidth=2.65&Measure_Suffix=%20mm
```

Accuracy

The plug-in does not support any perspective correction. This means that measuring scanned maps, diagrams will be accurate, measuring 3 dimensional objects is less accurate.

Plug-in Parameters

The following attributes can be assigned to the <plugin> tag. You might as well define all plug-in parameters via query or inside the <options> group by using the prefix "measure_" with the parameter name, e.g. "measure_ImageWidth"

MenuOffset	
Description	Indentation of the menu button
Syntax	Number
Default	0
Context	Plug-in attribute

Specifies the space in pixels left of the menu button of the plug-in.

ImageWidth	
Description	Real width of the entire image
Synta	Floating Point Value
Default	100.0
Context	Plug-in attribute

Defines the real width of the entire image (e.g. 120 inches). Please see the explanation in the previous section.

Prefix	
Description	Defines a prefix for the length value
Syntax	String
Default	---
Context	Plug-in attribute

Defines the prefix for the length value e.g. "length: ".

Suffix	
Description	Defines a suffix for the length value
Syntax	String
Default	mm
Context	Plug-in attribute

Defines the suffix for the length value e.g. "inches". Usually this is a length unit.

Decimals	
Description	Number of decimals of the length value
Syntax	Number
Default	1
Context	Plug-in attribute

Defines the number of decimals of the length value.

ShowText	
Description	Display the current measurement string
Syntax	Boolean
Default	true
Context	Plug-in attribute

Hides the text displaying the current measurement if set to "false".

ShowAngle	
Description	Display current angle
Syntax	Boolean
Default	true
Context	Plug-in attribute

If set to true the plug-in appends a text representing the current angle in degrees to the text display. E.g. "120.2 inches, 43.2°".

LineColor	
Description	Defines the color of the measure lines
Syntax	HexColor
Default	FF0000
Context	Plug-in attribute

6-digit hexadecimal color value defining the color of the measure lines.

TextColor	
Description	Defines the color of the text
Syntax	HexColor
Default	000000
Context	Plug-in attribute

6-digit hexadecimal color value defining the color of the text.

TextBorder	
Description	Defines the color of the text border
Syntax	HexColor or "false"
Default	000000
Context	Plug-in attribute
Version	4.1.0 or above

6-digit hexadecimal color value defining the color of the text border. Use "false" for no border.

TextSize	
Description	Size of the text
Syntax	Number (pixel)
Default	12
Context	Plug-in attribute

Text size for the textual output in pixels.

BGColor	
Description	Defines the color of the text background
Syntax	HexColor
Default	FFFFFF
Context	Plug-in attribute

6-digit hexadecimal color value defining the background color of the text.

BGAlpha	
Description	Defines the opacity of the text background
Syntax	Number
Default	80
Context	Plug-in attribute

Defines the opacity of the text background (0...100).

GlowRange	
Description	Size of the glow effect
Syntax	Number
Default	4
Context	Plug-in attribute

The glow range is a highlighted area surrounding the measure tool to improve its visibility. Using the GlowRange parameter you can define the size of the "glowing" area. Use "0" to disable the glow range.

GlowAlpha	
Description	Opacity of the glow range
Syntax	Number
Default	25
Context	Plug-in attribute

Opacity of the glow range surrounding the measure tool.

GlowColor	
Description	Color of the glow range
Syntax	HexColor
Default	FFFFFF
Context	Plug-in attribute

6-digit color defining the color of the glow range surrounding the measure tool.

Events	
Description	Send plug-in event notifications
Syntax	Boolean
Default	false
Context	Plug-in attribute
Version	3.2.5 and above

Specifies whether the plug-in sends plug-in event notifications. Plug-in events can be handled when using JSBridge plug-in or FSI Viewer Component.

The following events apply to this plug-in:

Event ID	Parameter	Description
onMeasureEnable	Boolean	The measure tool gets enabled or disabled.
onMeasureStart	String	The user starts measuring. The parameter provides the current distance and angle as displayed in the measure tool.
onMeasure	String	The user drags the measure tool. The parameter provides the current distance and angle as displayed in the measure tool.
onMeasureStop	String	The user stopped measuring. The parameter provides the current distance and angle as displayed in the measure tool.

12.17 Mousemodes



Plug-in Extends:

- FSI Viewer
- FSI Showcase

- FSIViewer JS
- FSI Showcase JS

Plug-in Location for Flash viewers:

/plugins/mousemodes.swf



Function:

Integrates buttons into the menu bar to select the mouse mode.

Syntax:

```
<Plugin src="mousemodes" />
```

The "Mousemodes" plug-in provides 3 buttons (2D images) or 4 buttons (3D presentations) to select the current mouse mode.

Plug-in Parameters

MenuOffset	
Description	Indentation of the menu buttons
Syntax	Number
Default	0
Context	Plug-in Tag

Specifies the space in pixels left of the menu button(s) of the plug-in.





Mode <i>n</i>	
Description	Removes the button for mouse mode <i>n</i> from the menu bar
Syntax	Boolean
Default	—
Context	Plug-in Tag

Provides the possibility to hide specific mouse mode buttons.

Remove Drag-Button

```
<Plugin src="mousemodes" Model="false" />
```

The example above shows how to hide the button for mouse mode 1 (Drag).

Mode n		Mouse Mode
Mode0		Zoom
Mode1		Pan
Mode2		Rotate 3D (X/Y Axis)
Mode3		Rotate 2D (Z Axis)

Sequence	
Description	Sequence of the buttons
Syntax	String
Default	0,1,3,2
Context	Plug-in Tag
Version	3.5.0 or higher

You can alter the sequence of the buttons by providing the modes separated by commas (see the table above). Example: "1,0,2,3" alters the sequence of the buttons to "Pan, Zoom, Rotate 3D, Rotate 2D".

12.18 MousemodeSelect



Plug-in Extends:

- FSI Viewer
- FSI Showcase
- FSI Pages
- FSI Pages mobile

Plug-in Location:

/plugins/mousemodeselect.swf

Function:

Integrates a button and a dropdown menu into the menu bar to select the mouse mode.



Syntax:

```
<Plugin src="mousemodeselect" />
```

The "Mousemode Select" plug-in adds a button and a dropdown menu to the menu bar to select the mouse mode.

Plug-in Parameters

MenuOffset	
Description	Indentation of the menu button
Syntax	Number
Default	0
Context	Plug-in Tag

Specifies the space in pixel left of the menu button of the plug-in.





Mode <i>n</i>	
Description	Removes the button for mouse mode <i>n</i> from the menu
Syntax	Boolean
Default	—
Context	Plug-in attributes

Provides the possibility to hide specific mouse mode buttons in the dropdown menu.

Remove Drag-Button

```
<Plugin src="mousemodes" Model="false" />
```

The example above shows how to hide the button for mouse mode 1 (Drag).

Mode n		Mouse Mode
Mode0		Zoom
Mode1		Pan
Mode2		Rotate 3D (X/Y Axis)
Mode3		Rotate 2D (Z Axis)

12.19 Music



Plug-in Extends:

- FSI Viewer
- FSI Showcase
- FSI Pages

Plug-in Location:

/plugins/music.swf

Function:

Adds streaming background music or sound to the FSI Viewer.

Syntax:

```
<Plugin src="backgroundimage"
FPXSrc="images/bgimage.tif" />
```



The "Music" plug-in adds a background sound or music to FSI Viewer. Additionally a button will be added to the menu bar to switch the music on or off.

Plug-in Parameters

MenuOffset	
Description	Indentation of the menu button
Syntax	Number
Default	0
Context	Plug-in attribute

Specifies the space in pixels left of the menu button of the plug-in.

AutoPlay	
Description	Start playing the sound on startup
Syntax	Boolean or String
Default	false
Context	Plug-in attribute

Set the value to Boolean "true" to make the sound play on startup.

From version 1.3 of the plug-in you can additionally set the value to "onLoad" to start the sound after image data has been loaded completely.

Song	
Description	Path to an MPEG3 audio file (.mp3)
Syntax	URL
Default	—
Context	Plug-in attribute

Defines an absolute or relative path to an MPEG3 file (*.mp3). If you specify a relative path (instead "http://..." or "https://...") the path needs to be provided relative to the "music" subdirectory of the "/fsi" folder.

Loop	
Description	Loop / Play the sound once
Syntax	Bool
Default	false
Context	Plug-in attribute

Defines if the sound will be played once or in a continuous loop.

PauseOnStop	
Description	Pause music rather than stop when clicking the plug-in button
Syntax	Boolean
Default	false
Context	Plug-in attribute
Version:	4.0.7

By default pressing the plug-in button stops/plays the music. Setting this parameter to "true" makes the plug-in pause/resume when clicking the plug-in button so that the song does not restart from the beginning.

Volume	
Description	Sound volume
Syntax	Number (0-100)
Default	—
Context	Plug-in attribute

Defines the volume of the sound from 0 (quiet) to 100 (maximum).

KeepState	
Description	Keep Stop/Play state when selecting images
Syntax	Boolean
Default	true
Context	Plug-in attribute

For viewers with multiple images only.

Keep the playing state and ignore the "AutoPlay" parameter when switching from one image to another. This way "AutoPlay=true" does not start the sound if the user switched the music off.

12.20 NavExtension



Plug-in Extends:

- FSI Viewer
- FSI Showcase

Plug-in Location:

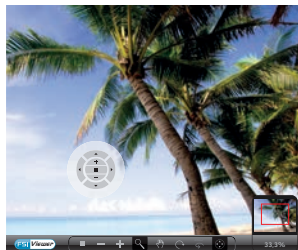
/plugins/navextension.swf

Function:

Additional navigation control.

Syntax:

```
<Plugin src="navextension"
visible="true" />
```



The "NavExtension" plug-in adds a draggable navigation control to the FSI Viewer user interface that can be hidden and shown with a corresponding button in the menu bar. The control automatically adds buttons for scenes and scene sets depending on the image configuration.

Plug-in Parameters

MenuOffset	
Description	Indentation of the menu button
Syntax	Number
Default	0
Context	Plug-in attributes

Specifies the space in pixels left of the menu button of the plug-in.

Visible	
Description	Initial display state
Syntax	Boolean
Default	false
Context	Plug-in attribute

Defines the initial visibility state of the control.

X	
Description	Initial X position in pixel
Syntax	Number
Default	0
Context	Plug-in Tag

Initial horizontal position of the control in pixels relative to the top-left corner of the viewer.

Y	
Description	Initial Y position in pixel
Syntax	Number
Default	0
Context	Plug-in Tag

Initial vertical position of the control in pixels relative to the top-left corner of the viewer.

12.21 Notepad



Plug-in Extends:

- FSI Viewer
- FSI Showcase
- FSI Pages

Plug-in Location:

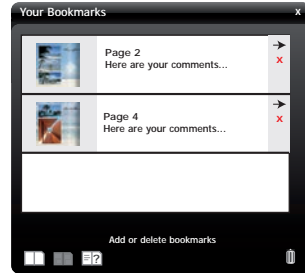
/plugins/notepad.swf

Function:

Creating and displaying bookmarks with text and image.

Syntax:

```
<Plugin src="notepad" />
```



The notepad plug-in enables the user to create bookmarks for an image section with descriptive text. The available bookmark buttons and options depend on the Add-on you are using:

Add-on	Description
FSI Pages	<ul style="list-style-type: none"> - Bookmark current page(s) - Bookmark product link * - Bookmark image section
FSI Showcase	<ul style="list-style-type: none"> - Bookmark current image - Bookmark image section of the current image
None	<ul style="list-style-type: none"> - Bookmark entire image - Bookmark image section

* for bookmarks containing descriptive text (<text> nodes in XML page data, e.g. for the LargeToolTip plug-in) this text is being used as default text for the bookmark (see image above)

Unique ID

It's crucial to specify a unique ID for each application of the notepad plug-in in order to store the notepad data in a separate data set on the user's system. Otherwise the user might see bookmarks for another image or catalog. You can either use the global FSI Viewer parameter "UniqueID" or use the Notepad plug-in parameter. Please refer to the "UniqueID" parameter description below for details.

Persistent Data Storage

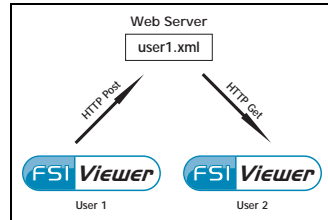
The notepad uses an internal Flash technique to permanently store the notepad data on the users system. No server side script or cookies are required.

Please note that the notepad data does not store the thumbnail images, but the URL to the thumbnail instead. If you remove an image from your imaging server the user will not see the thumbnail any more when opening the notepad later on.

Posting and Loading data

The notepad plug-in provides the functionality to post bookmarks in XML format to a web server. Additionally the plug-in is capable of reading an XML data file from the server.

Using a little server side script you can this way setup a number of interesting communication systems. For example a "send bookmarks by email" option.



In this case you store the XML data file received from "User1's" notepad plug-in on a web server and email a link to an FSI Viewer instance including the name of the stored XML file returned by the web server to User2. This way User2 can see the bookmarks and remarks created by User1.

Additionally you can provide predefined notepad data, for example to present hot products, special offers and alike. In this case you provide the URL to a predefined XML data file using the "LoadXML" parameter. This way the notepad on the client computer loads the notepad data from this file instead of using any data stored locally on the user's computer.

Plug-in Parameters - Basic



Required Parameter!

UniqueID	
Description	Unique String used to store notepad data locally
Syntax	String (see description)
Default	---
Context	Plug-in attribute

A unique string identifying the notepad data. Using the same (or no) UniqueID for different images or image collections (catalogs) will present false bookmarks in the notepad. You can for example use the path of the image or the name of a catalog. Do not use the following characters in UniqueIDs: [Space] ~ % & \ ; : " ' , < > ? # Alternatively to the plug-in parameter you can use the global FSI Parameter "UniqueID" to specify a unique identifier.

Revision	
Description	Revision of the notepad data
Syntax	Number
Default	1
Context	Plug-in attribute

You can use different revisions for the same UniqueID. Changing the revision invalidates all bookmarks for a given UniqueID. You might for example change the revision if you change, add or remove pages in an FSI Pages catalog.

Plug-in Parameters – Layout and Appearance

Visible	
Description	Show or hide the notepad on startup
Syntax	Boolean
Default	false
Context	Plug-in attribute

Show (true) or hide (false) the notepad on startup.

Width, Height	
Description	Width and height of the notepad window
Syntax	Number (pixels)
Default	320 (width) and 272 (height)
Context	Plug-in attribute

Width and height in pixels of the notepad window.

ImageWidth, ImageHeight	
Description	Max. width and height of thumbnail images
Syntax	Number (pixels)
Default	80 (width) and 66 (height)
Context	Plug-in attribute

The maximum width and height of thumbnail images in pixels.

BackgroundColor	
Description	Background color of the notepad window.
Syntax	HexColor(s)
Default	FFFFFF,DDDDDD
Context	Plug-in attribute
Version	Obsolete from version 5.0.0 (defined by FSI Viewer Skin)

Background color of the notepad window. You can specify a solid color or use multiple hexadecimal color values concatenated by, " to specify a color gradient, e.g. "FF0000,00FF00,FF0000".

BackgroundGradientAngle	
Description	Angle of the background color gradient
Syntax	Number (degrees)
Default	45
Context	Plug-in attribute
Version	Obsolete from version 5.0.0 (defined by FSI Viewer Skin)

Angle of the background color gradient in degree. The default value 45 creates a gradient from top left to bottom right. Has no effect if you used a single color value for the BackgroundColor parameter.

FrameWidth	
Description	Width of the window frame
Syntax	Floating Point Number (pixels)
Default	1.1
Context	Plug-in attribute
Version	Obsolete from version 5.0.0 (defined by FSI Viewer Skin)

Width of the window border.

FrameColor	
Description	Color of the window frame
Syntax	HexColor
Default	000000
Context	Plug-in attribute
Version	Obsolete from version 5.0.0 (defined by FSI Viewer Skin)

Color of the window border.

EdgeRadius	
Description	Radius of the window corners
Syntax	Number
Default	6
Context	Plug-in attribute
Version	Obsolete from version 5.0.0 (defined by FSI Viewer Skin)

Radius of the window corners. Use "0" for straight edges.

ImageBackgroundColor	
Description	Color of the background behind the thumbnails
Syntax	HexColor
Default	EEEEEE
Context	Plug-in attribute

Color of the background behind the thumbnails.

ItemButtonsBackgroundColor	
Description	Color of the background behind the item buttons
Syntax	HexColor
Default	EEEEEE
Context	Plug-in attribute

Color of the background behind the items buttons right of the bookmarks.

TextColor	
Description	Color of the hints (tool tips) displayed below the list
Syntax	HexColor
Default	000000
Context	Plug-in attribute

Color of the help texts below the bookmark list.

ScrollBaseColor, ScrollArrow, ScrollTrack	
Description	Colors for the scroll bar
Syntax	HexColor
Default	---
Context	Plug-in attribute

ScrollBaseColor defines the base color for the scroll bar in the bookmark list. Additionally you specify a HexColor value for the Arrows and the track area of the scroll bar.

EnableAddFullBookmark, EnableAddLinkBookmark, EnableAddCustomBookmark, EnableSendBookmarks	
Description	Show or hide the buttons
Syntax	Boolean
Default	Depending on loaded add-on (SendBookmarks: false)
Context	Plug-in attribute

Use these parameters to enable or disable the buttons below the bookmark list.

EnableAddFullBookmark:	Enable bookmarking of entire images or pages
EnableAddLinkBookmark:	Enable bookmarking of links
EnableAddCustomBookmark:	Enable bookmarking of custom areas
EnableSendBookmarks:	Enable sending bookmark data

ImageEffects	
Description	Image modification parameters
Syntax	String
Default	---
Context	Plug-in attribute

Optional parameter to add image parameters for the thumbnail images. For FSI Server you can for example sharpen an image or adjust the image compression level, e.g. "effects=sharpen(230)&quality=95".

ExpandSelectionToImage	
Description	Expand image selections to fill entire thumbnail
Syntax	Boolean
Default	true
Context	Plug-in attribute

By default the notepad plug-in expands bookmark selections (custom and link bookmarks) to fill the entire thumbnail area. Setting the value to "false" disables expanding bookmark selections.

AddButton	
Description	Add notepad button to menu bar
Syntax	Boolean
Default	true (FSI Pages: false)
Context	Plug-in attribute

Add a button to the FSI Viewer menu bar to show or hide the notepad. FSI Pages skins provide a button independent of this parameter.

MenuOffset	
Description	Offset of the notepad button
Syntax	Number (pixels)
Default	0
Context	Plug-in attribute

Offset of the notepad button in the menu bar (FSI Viewer only).

FlipAndZoomInOneGo	
Description	Flip and zoom in one go after clicking an FSI Pages bookmark
Syntax	Boolean
Default	true
Context	plug-in node or HTTP query prefixed with "notepad_"
Version	5.6.0

Setting this parameter to "false" changes the behavior when clicking an FSI Pages bookmark containing a bookmark area. The first click flips to the bookmarked page, the second click zooms the page and the third click zooms to the bookmarked area. The actual number of clicks depend on the page you are viewing and if you already zoomed the bookmarked page.

Setting this parameter to "true" (default) the actions flip (and zoom) will be execute in one go requiring one click only.

Plug-in Parameters – Loading and Posting Data

The parameters below are required only if you want to send or load bookmark data to or from a web server.

Loading bookmarks from an XML file is as easy as specifying the URL to the XML data file using the parameter "LoadXML". You can use the "LoadXMLPrefix" parameter to prefix all "LoadXML" parameters, e.g. with the server URL.

Sending the XML data to a web server:

- Make sure to enable the "EnableSendBookmarks" parameter to display the corresponding button below the bookmark list

- If the user clicks the button, the plug-in sends the notepad data in XML format via HTTP POST to the server. The data is being send in the "data" post variable
- The answer (the content of the receiving page) of the web server in return must be simple XML data:

```
<result success="true">filename.xml</result>
```

 if the server succeeded in saving the file OR

```
<result success="false">error reason</result>
```

Depending on the result the plug-in displays the web page specified by the "SendXMLUriSuccess" or "SendXMLUriFailed".

The plug-in posts the following values to the success or failure HTML page:

Data	Description
datafile	The file name of the stored XML file returned by the server
error	Error description
FSI_Url	The URL of the calling FSI Viewer instance
FSI_notepad_id	The unique id of the notepad (UniqueID)

LoadXML	
Description	URL or file name of an XML data file to load
Syntax	String
Default	---
Context	Plug-in attribute

URL or file name of an XML data file to load from a web server. If you add this parameter, the notepad loads the bookmarks from the file specified instead of loading bookmarks from the user's system.

LoadXMLPrefix	
Description	Prefix to use for the LoadXML parameter
Syntax	String
Default	---
Context	Plug-in attribute

You can use this value e.g. in the `_default.xml` configuration file to specify the server address and folder to load XML notepad data files from.

Example: "http://foo.com/notepaddata/".

SendXMLStoreURL	
Description	URL to post notepad data to
Syntax	String
Default	---
Context	Plug-in attribute

URL of a web server receiving the notepad data in XML format. See "Loading and Posting Data" on the preceding page for details.

SendXMLUriSuccess	
Description	URL to open if sending XML data succeeded
Syntax	String
Default	---
Context	Plug-in attribute

Url to open if sending the XML data to the web server specified by "SendXMLStoreURL" parameter succeeded.

SendXMLUriFailed	
Description	URL to open if sending XML data failed
Syntax	String
Default	---
Context	Plug-in attribute

URL to open if sending the XML data to the web server specified by "SendXMLStoreURL" parameter failed.

SendXMLUriSuccessTarget, SendXMLUriFailedTarget	
Description	HTML target frame to open the URL in
Syntax	String
Default	_blank
Context	Plug-in attribute

HTML target frame to open the "SendXMLUriSuccess" or the "SendXMLUriFailed" page in.

Plug-in Parameters – Advanced Customization Customizing Image

You can replace the default animations / images displayed when the bookmark list is empty, when the user adds a custom or a link bookmark.

The plug-in loads the optional custom help images from your imaging server. The syntax of these parameters therefore corresponds to the "FPXSrc" parameter. If you specified the "FPXBase" parameter you only need to provide the path to the image, e.g. "helpimages/emptynotepad.tif".

The following images can be defined:

ImageHelpMain:	image to display when the bookmark list is empty
ImageHelpLinkBookmark:	image to display while the user adds a link bookmark in FSI Pages
ImageHelpLinkCustom:	image to display while the user adds a custom bookmark

12.22 PagesLibrary



Plug-in Extends:

- FSI Pages mobile

Plug-in Location:

/plugins/pageslibrary.swf

Function:

Provides access to a list of additional publications to the user.



Syntax:

```
<Plugin src="pageslibrary" datafile="library.xml" />
```

Display a list of former publications to the user to choose from. The list of publications is defined using XML in the specified datafile. The value of the parameter datafile needs to be pageslibrary in order for the tag to work.

Plug-in Parameters

DataFile	
Description	URL to an external XML file describing the library structure
Syntax	String
Default	—
Context	Plug-in tag

Relative URL to an external XML file to provide the XML data required for this plug-in. The XML file must be located in the same domain as FSI Viewer to avoid cross-domain security restrictions.

The XML file contains a list of items to display in the library dialog. Each item must contain the following attributes:

src = the path to the images for this publication. The first image will be used to display the thumbnail.

label = optional description of the publication

cfg = the configuration file to open if the user clicks the thumbnail

See "Library XML data structure" below for details and an example.

ImageWidth	
Description	defines width of the cover image in the dialog window
Syntax	Number
Default	200
Context	Plug-in tag

This value defines the width of the cover image in the dialog window.

ImageHeight	
Description	defines height of the cover image in the dialog window
Syntax	Number
Default	120
Context	Plug-in tag

This value defines the height of the cover image in the dialog window.

AddButton	
Description	defines whether the plug-in icon is displayed
Syntax	Boolean
Default	false if FSI Pages is activated, otherwise true
Context	Plug-in tag

Defines whether the plug-in icon is displayed in the FSI Viewer toolbar. Defaults to FALSE if FSI Pages is activated.

MenuOffset	
Description	defines offset of all custom buttons in pixels to elements of menu bar.
Syntax	Number
Default	0
Context	Plug-in tag

Defines an offset of all custom buttons in pixels to elements of the menu bar.

MarginX/MarginY	
Description	define the margin in pixels of the the content in the dialog window
Syntax	Number
Default	12
Context	Plug-in tag

These values define the margin in pixels of the the content in the dialog window.

ShowLabels	
Description	defines whether the service labels should be displayed
Syntax	Boolean
Default	false
Context	Plug-in tag

This value defines whether the service labels should be displayed.

DialogWidth	
Description	defines the width of the dialog window
Syntax	Number
Default	480
Context	Plug-in tag

Defines the width of the dialog window.

DialogHeight	
Description	defines the height of the dialog window
Syntax	Number
Default	430
Context	Plug-in tag

Defines the height of the dialog window.

Library XML data structure

Each publication item is defined in an <item> node. Three node attributes defines the publication source, label and fsi configuration file:

src = Image Path on Server

label = Title of publication to be displayed

cfg = Configuration file to be used

XML Sample
<pre><?xml version="1.0" encoding="UTF-8" ?> <library> <item src="/demo/January/" label="Issue 01" cfg="cfg_01" /> <item src="/demo/February/" label="issue 02" cfg="cfg_02" /> </library></pre>



12.23 PagesMirror

Plug-in Extends:

- FSI Pages

Plug-in Location:

/plugins/pagesmirror.swf

Function:

Displays a mirror like effect.

Syntax:

```
<Plugin src="pagesmirror"/>
```



Plug-in Parameters

Blur	
Description	Defines the blur effect of pages
Syntax	Number
Default	---
Context	PagesMirror
Version	5.0.0

Defines the blur factor of the mirrored images.

Offset	
Description	Defines distance between the pages and the mirror effect
Syntax	Number (Pixels)
Default	---
Context	PagesMirror
Version	5.0.0

Defines the distance of the mirror effect and the pages.

StartAlpha	
Description	Alphablending
Syntax	Number
Default	70
Context	PagesMirror
Version	5.0.0

Defines the mirror effects start opacity .

EndAlpha	
Description	Alphablending
Syntax	Number
Default	0
Context	PagesMirror
Version	5.0.0

Defines the mirror effects opacity on the near end.



Please note: For technical reasons the PagesMirror plugin will not work when using page overlays

12.24 PageSounds



Plug-in Extends:

- FSI Pages

Plug-in Location:

Function:

Adds page flip sounds to FSI Pages

Syntax:

```
<Plugin src="pagesounds" />
```

This plug-in adds page flip sounds to FSI Pages and does not have any effect with FSI Showcase or FSI Viewer instances.

Plug-in Parameters

Volume	
Description	Volume of the page flip sounds
Syntax	Number
Default	50
Context	Plug-in tag

The volume to play the flip sounds at from "0" (off) to "100" (maximum volume).

Library	
Description	File name of the sound library to use
Syntax	File Name
Default	pagesoundlib_coated.swf
Context	Plug-in tag

File name of an SWF file (compiled Flash movie) containing the flip sounds. FSI Viewer ships with two different sound libraries:

- pagesoundlib_coated.swf (coated pages like catalogs)
 pagesoundlib_newspaper.swf (uncoated pages like newspapers)



12.25 PagesThumbBar



Plug-in Extends:

- FSI Pages

Plug-in Location:

/plugins/thumbbar.swf

Function:

Displays a bar with thumbnail images

Syntax:

```
<Plugin src="thumbbar"/>
```



Plug-in Parameters

AlignLeft	
Description	Keep current image aligned to the left
Syntax	Number (Pixels)
Default	false
Context	PagesThumbBar
Version	5.0.0

When enabled, the thumbnail of the currently active image is shown left in the thumbnail bar. When disabled, the thumbnail is shown in the center below the catalog.

Arrows	
Description	Display scroll arrows
Syntax	Boolean
Default	true
Context	PagesThumbBar
Version	5.0.4

If set to false, the arrows for scrolling the thumbnailbar are hidden and the user needs to drag the thumbs to scroll.

ArrowAlphaActive	
Description	Opacity of arrow
Syntax	Number
Default	100
Context	PagesThumbBar
Version	5.0.0

Opacity of the arrows when mouse hovers over the thumbnail bar.

ArrowAlphaNormal	
Description	Opacity of arrow
Syntax	Number
Default	0
Context	PagesThumbBar
Version	5.0.0

Defines the opacity of the arrows in normal state.

ArrowFillColor	
Description	Arrow fill color
Syntax	HexColor
Default	FFFFFF
Context	PagesThumbBar
Version	5.0.0

The fill color of the arrows inside in the thumbnailbar.

ArrowLineColor	
Description	Arrow line color
Syntax	HexColor
Default	FFFFFF
Context	PagesThumbBar
Version	5.0.0

The fill color of the arrows inside in the thumbnailbar.

DropShadow	
Description	Enable Thumbnail Shadows
Syntax	Boolean
Default	false
Context	PagesThumbBar
Version	5.0.0

Enable/Disable the drop shadow effect of the thumbnails.

Height	
Description	Size of thumbnail images
Syntax	Number (Pixels)
Default	80
Context	PagesThumbBar
Version	5.0.0

Defines the height of the thumbnails in the thumbnail bar. The width is derived from the image size.

HideOnZoom	
Description	Hide thumbnails when zooming
Syntax	Boolean
Default	false
Context	PagesThumbBar
Version	5.0.0

When enabled, the thumbs will be hidden if an image is zoomed into.

MagnifyCurrentPages	
Description	Size of thumbnail images
Syntax	Float
Default	100
Context	PagesThumbBar
Version	5.0.0

Defines the magnification level of the current thumbnail. Values greater than 100 will increase, smaller values will decrease the thumbnail size.

MarginBottom	
Description	Margin of the thumb bar
Syntax	Number (Pixels)
Default	0
Context	PagesThumbBar
Version	5.0.0

Defines the bottom margin of the thumbnail bar.

MarginTop	
Description	Margin of the thumb bar
Syntax	Number (Pixels)
Default	8
Context	PagesThumbBar
Version	5.0.0

Defines the top margin of the thumbnail bar.

OffsetLeft	
Description	Offset of thumbnail bar
Syntax	Number (Pixels)
Default	0
Context	PagesThumbBar
Version	5.0.0

Defines the offset of the thumbnail bar.

OffsetRight	
Description	Offset of thumbnail bar
Syntax	Number (Pixels)
Default	0
Context	PagesThumbBar
Version	5.0.0

Defines the offset of the thumbnail bar.

Padding	
Description	Thumbnail Padding
Syntax	Number (Pixels)
Default	10
Context	PagesThumbBar
Version	5.0.0

Defines the distance between the thumbnails.

SelectionBorderWidth	
Description	Border of the active thumbnail
Syntax	Number
Default	4
Context	PagesThumbBar
Version	5.0.4

Width of the border around the active thumbnail.

SelectionColor	
Description	Color of the active thumbnail
Syntax	HexColor
Context	PagesThumbBar
Version	5.0.4

Defines the backgroundcolor of the active pages in the thumbnail bar.

SelectionMarginBottom	
Description	Bottom margin of the active thumbnail
Syntax	Number
Default	6
Context	PagesThumbBar
Version	5.0.4

Defines the bottom margin of the active thumbnail.

SelectionRoundCorner	
Description	Round corner size of active thumbnail
Syntax	Number
Default	4
Context	PagesThumbBar
Version	5.0.4

Defines the rounded corner effect for the active thumbnail. Straight corners if set to 0.

TextColor	
Description	Color of page numbers
Syntax	HexColor
Context	PagesThumbBar
Version	5.0.4

Defines the rounded corner effect for the active thumbnail. Straight corners if set to 0. Defines the color of the page numbers in the Thumbbar Plugin.

TextColorHighlight	
Description	Round corner size of active thumbnail
Syntax	Number
Context	PagesThumbBar
Version	5.0.4

Defines the textcolor of the active thumbnail.

12.26 PrintSave



Plug-in Extends:

- FSI Viewer
- FSI Showcase

Plug-in Location:

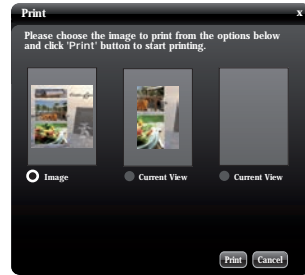
/plugins/printsave.swf

Function:

Add printing and saving capabilities to FSI Viewer.

Syntax:

```
<Plugin src="printsave"/>
```



Plug-in Parameters

MenuOffset	
Description	Offset in pixel to previous menu elements
Syntax	Number (Pixels)
Default	---
Context	PrintSave
Version	5.0.0

Offset in pixel to previous menu elements.

Print	
Description	Enable/Disable printing
Syntax	Boolean
Default	---
Context	PrintSave
Version	5.0.0

If disabled, the menu button is hidden.

PrintEffects	
Description	Image effects
Syntax	String (urlencoded)
Default	---
Context	PrintSave
Version	5.0.0

Defines image modification parameters to apply to images used for printing. The value of this parameter depends on the imaging server being used. With FSI Server you can sharpen the image and define the JPEG compression level, e.g. "effects=sharpen(230)&quality=95".

PrintResolution	
Description	Max. size of image to print
Syntax	Number (Pixels)
Default	2000
Context	PrintSave
Version	5.0.0

Defines the maximum size of the image in pixels to download for printing. Please note that the actual size depends on the size of the image and (with FSI server) on the maximum size defined in the real time template used with FSI Viewer (see parameter PrintTemplate).

PrintTemplate	
Description	Image template to be used
Syntax	String
Default	---
Context	PrintSave
Version	5.0.0

Defines the FSI Server profile used to load the print image.

Save	
Description	Enable/Disable saving
Syntax	Boolean
Default	true
Context	PrintSave
Version	5.0.0

If disabled, the menu button is hidden.

SaveEffects	
Description	Image effects
Syntax	String (urlencoded)
Default	---
Context	PrintSave
Version	5.0.0

Defines image modification parameters to apply to images used for saving. The value of this parameter depends on the imaging server being used. With FSI server you can sharpen the image and define the JPEG compression level, e.g. "effects=sharpen(230)&quality=95".

SaveResolution	
Description	Max. size of image to save
Syntax	Number (Pixels)
Default	600
Context	PrintSave
Version	5.0.0

Defines the maximum size of the image in pixels to download for printing. Please note that the actual size depends on the size of the image and (with FSI server) on the maximum size defined in the real time template used with FSI Viewer (see parameter PrintTemplate).

SaveTemplate	
Description	Image template to be used
Syntax	String
Default	---
Context	PrintSave
Version	5.0.0

Defines the FSI Server profile used to download an image.

12.27 Resize



Plug-in Extends:

- FSI Viewer
- FSI Showcase
- FSI Pages
- FSI Pages mobilea

Plug-in Location:

/plugins/resize.swf

Function:

Rearranges user interface elements after resizing the viewer

Syntax:

```
<Plugin src="resize" MaxWidth="800
MaxHeight="600"/>
```



Using the resize plug-in you can define the viewer dimension in percent of your HTML document. After resizing the plug-in rearranges the user interface according to the size of the object tag without reloading the entire viewer instance and without the need for client side scripting. Note that this plug-in is required only, if you want to allow the user to dynamically resize a FSI Viewer instance.

Plug-in Parameters

MaxWidth and MaxHeight	
Description	Maximum width and height
Syntax	Number in pixel
Default	—
Context	Plug-in Tag

FSI Viewer can not be resized to a dimension greater than specified by these values.

MinWidth and MinHeight	
Description	Minimum width and height
Syntax	Number in pixel
Default	—
Context	Plug-in Tag

FSI Viewer can not be resized to a dimension smaller than specified by these values.

EnlargeBy	
Description	Resize viewer in steps of n pixels
Syntax	Number
Default	1
Context	Plug-in tag

Specifies the steps FSI Viewer will be resized by. Entering 1 means that the viewer resizes exactly according to the size of the object tag. Entering greater numbers forces the viewer to resize by the number of pixels specified. Using values > 1 increases the chance of imaging server cache hits and might thus reduce server load.

12.28 SelectFrame



Plug-in Extends:

- FSI Viewer
- FSI Showcase

Plug-in Location:

/plugins/selectframe.swf

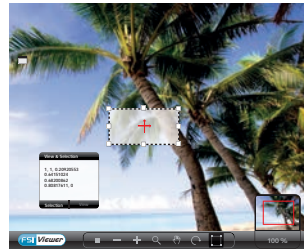
Function:

Provides a resizable select frame

Syntax:

```
<Plugin src="selectframe" />
```

This plug-in adds a resizable select frame into FSI Viewer.



Pressing SHIFT locks the aspect ratio of the selection.

Pressing CTRL resizes the selection relative to the center

Pressing ALT rotates the selection

Pressing ALT + SHIFT rotates the selection in steps of 5 degrees

Plug-in Parameters

The following attributes can be assigned to the <plugin> tag.

You might as well define all plug-in parameters via query or inside the <options> group by using the prefix "selectframe_" with the parameter name, e.g. "selectframe_window".

MenuOffset	
Description	Indentation of the menu button
Syntax	Number
Default	0
Context	Plug-in attribute

Specifies the space in pixels left of the menu button of the plug-in.

Visible	
Description	Show or hide the plug-in on startup
Syntax	Boolean
Default	false
Context	Plug-in attribute

Show or hide the select box on startup.

Window	
Description	Show or hide the output window
Syntax	Boolean
Default	true
Context	Plug-in attribute

Show or hide the output window of the plug-in.

AnimatedRect	
Description	Enable or Disable frame animation
Syntax	Boolean
Default	false
Context	Plug-in attribute

If enabled the frameborders will be animated.

Rotate	
Description	Enable rotation of the selection
Syntax	Boolean
Default	true
Context	Plug-in attribute

Enable rotating the selection by pressing ALT key or ALT + SHIFT keys and dragging the handles of the select box.

MaskColor	
Description	Color of the mask
Syntax	HexColor
Default	FFFFFF
Context	Plug-in attribute

Define a 6-digit hexadecimal color value for the mask of the select box.

MaskAlpha	
Description	Opacity of the mask
Syntax	Number
Default	75
Context	Plug-in attribute

Defines the opacity of the mask of the select box (0 = invisible).

AspectRatio	
Description	Aspect ratio of the selection
Syntax	String
Default	---
Context	Plug-in attribute

Defines an aspect ratio for the selection, e.g. "320:200". Use SHIFT to lock or unlock the aspect ratio when resizing the selection.

InvertShift	
Description	Inverts the function of the SHIFT key
Syntax	Boolean
Default	false
Context	Plug-in attribute

You can use this parameter to invert the function of the SHIFT key while dragging.

Callback	
Description	JavaScript callback function
Syntax	String
Default	---
Context	Plug-in attribute

Specify the name of a custom JavaScript function to call each time the selection changes. The arguments passed to this function are:

SceneSet, Scene, Left, Top, Right, Bottom, Rotation, Current View

Example: JavaScript callback function

```
function onSelect (set, scene, l, t, r, b, rot, strView) {
  doSomething();
}
```

Additionally you can use the template "\$\$selection\$\$" for the
→ *CustomButton* plug-in.

Events	
Description	Send plug-in event notifications
Syntax	Boolean
Default	false
Context	Plug-in attribute
Version	3.2.5 and above

Specifies whether the plug-in send plug-in event notifications. Plug-in events can be handled when using JSBridge plug-in or FSI Component.

The following events apply to this plug-in:

Event ID	Parameter	Description
onSelectFrameEnable	Boolean	The select has been enabled or disabled.
onSelectFrameStart	---	The user starts selecting.
onSelectFrame	String	The user modifies the select frame. The parameter provides a string representing the current selection.
onSelectFrameStop	String	The user stopped modifying the selection. The parameter provides a string representing the current selection.

12.29 ShoppingList



Plug-in Extends:

- FSI Pages

Plug-in Location:

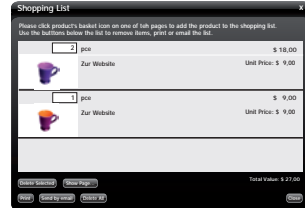
/plugins/shoppinglist.swf

Function:

Add shopping cart functionality to FSI Pages.

Syntax:

```
<Plugin src="shoppinglist"/>
```



Please note: This Plug-in requires the Template Sever.

Plug-in Parameters

Currency	
Description	Defines the currency
Default	---
Context	ShoppingList
Version	5.0.0

Defines the currency inside the shoppinglist.

DecimalCharacter	
Description	Decimal delimiter
Syntax	String
Default	---
Context	ShoppingList
Version	5.0.0

Decimal delimiter.

DefaultUnit	
Description	Default unit string
Syntax	String
Default	pce
Context	ShoppingList
Version	5.0.0

Defines the default unit of items added to the shoppinglist.

Email	
Description	Enable/Disable sending list by email
Default	true
Context	ShoppingList
Version	5.0.0

When enabled, the user can email a shoppinglist to others.

MaxCountPerItem	
Description	Maximum Count per item
Syntax	Number
Default	99
Context	ShoppingList
Version	5.0.0

Defines the maximum unit per item that can be added to the shoppinglist.

MinItemHeight	
Description	Minimum Height of items in the list
Syntax	Number
Default	64
Context	ShoppingList
Version	5.0.0

Defines the minimum height of each item in the list.

PersistentStorage	
Description	Store shopping list data on the user's computer
Default	true
Context	ShoppingList
Version	5.0.0

Enable or disable the caching of list data. If disabled, the list info is lost when the user leaves the FSI pages instance.

PersistentStorageExpiresAfter	
Description	Expire list data
Syntax	Number
Context	ShoppingList
Version	5.0.0

Defines after which period of time the shopping list data expires.

Print	
Description	Minimum Height of items in the list
Default	true
Context	ShoppingList
Version	5.0.0

Enable/Disable printing.

ShowListOnAddItem	
Description	Show shoppinglist behaviour
Syntax	string
Default	once
Context	ShoppingList
Version	5.0.0

Defines whether the shoppinglist is displayed after an item has been added.

TargetURL	
Description	Defines the target for shoppinglist
Default	---
Context	ShoppingList
Version	5.0.0

Defines the target URL of the servlet the shoppinglist data is send to.

Template	
Description	Template to use
Syntax	template name
Context	ShoppingList
Version	5.0.0

Defines the HTML templates to be used for rendering the pages.

DialogHeight	
Description	Defines the height of the shoppinglist dialog window
Syntax	Number
Default	400
Context	ShoppingList
Version	5.0.0

Defines the height of the shoppinglist dialog window.

DialogWidth	
Description	Defines the width of the shoppinglist dialog window
Syntax	Number
Default	600
Context	ShoppingList
Version	5.0.0

Defines the width of the shoppinglist dialog window.

12.30 SoftwareCursor

Plug-in Extends:

- FSI Viewer
- FSI Showcase
- FSI Pages

Plug-in Location:

/plugins/softwarecursor.swf

Function:

Adds a software cursor to the FSI Viewer.

Syntax:

```
<Plugin src="softwarecursor" />
```

The "Software Cursor" plug-in adds a software cursor representing the current mouse mode (Zoom, Drag, Rotate-2D, Rotate 3D).

Plug-in Parameters

Library	
Description	File name of an SWF containing cursor graphics
Syntax	String
Default	— (built-in library)
Context	Plug-in attributes

File name of a SWF (compiled Flash movie) file containing custom cursor graphics.



12.31 StickyNotes



Plug-in Extends:

- FSI Pages

Plug-in Location:

/plugins/stickynotes.swf

Function:

Adds bookmark functionality to FSI Pages.

Syntax:

```
<Plugin src="stickynotes"/>
```

Plug-in Parameters



AllowMirroredNotes	
Description	Enable/Disable mirrored note text
Syntax	Boolean
Default	false
Context	StickyNotes
Version	5.0.0

When enabled, the notes text will be mirrored when the page is turned, otherwise it will not be displayed.

Alpha	
Description	Opacity of the sticky notes
Syntax	Number
Default	50
Context	StickyNotes
Version	5.0.0

Defines the opacity of the sticky notes from invisible (0) to opaque (100).

Color	
Description	Defines the color of the sticky notes
Syntax	Color
Default	FFFF55
Context	StickyNotes
Version	5.0.0

Defines the main color of the sticky notes.

TextColor	
Description	Defines the color of the page numbers
Syntax	Color
Default	000000
Context	StickyNotes
Version	5.5.3

Color of the page numbers on sticky notes.

ColorHighlight	
Description	Color of lighter part
Syntax	Color
Default	FFFEE
Context	StickyNotes
Version	5.0.0

Defines the color of the lighter part of the sticky notes.

ColorShadow	
Description	Defines the color of the dark part of the sticky notes
Syntax	Color
Default	CCCC00
Context	StickyNotes
Version	5.0.0

Defines the color of the dark part of the sticky notes.

NaturalRotation	
Description	Enables slightly rotated sticky notes
Syntax	Boolean
Default	true
Context	StickyNotes
Version	5.0.0

When this option is enabled, the sticky notes will be rotated slightly, to give a more natural look and feel.

PageNumbers	
Description	Enable/Disable Page numbers
Default	True
Context	StickyNotes
Version	5.0.0

Display page numbers on the sticky notes.

PersistentStorage	
Description	Cache notes information
Default	True
Context	StickyNotes
Version	5.0.0

When disabled, the sticky notes will be deleted when the user leaves the FSI Pages instance. When enabled the data will be cached for the given time (defined by PersistentStorageExpiresAfter).

PersistentStorageExpiresAfter	
Description	Keep note data for given time
Syntax	Number
Context	StickyNotes
Version	5.0.0

Defines after which period of time the cached note data expires.

Position	
Description	Position of the sticky notes
Syntax	String
Default	top
Context	StickyNotes
Version	5.0.0

Sticky notes position on the page (top, bottom or right).

Scale	
Description	Size of the sticky notes
Syntax	Number
Default	100
Context	StickyNotes
Version	5.0.0

Defines the size of the sticky notes.

Spacing	
Description	Defines the distance between the sticky notes
Syntax	Number (Pixels)
Default	8
Context	StickyNotes
Version	5.0.0

Defines the distance between the sticky notes.

CornerRadius	
Description	Defines corner radius of the sticky notes
Syntax	Float
Default	4
Context	StickyNotes
Version	5.5.3

Defines the corner radius of the sticky notes. Use 0 for a squared shape.

12.32 Synchronize



Plug-in Extends:

- FSI Viewer
- FSI Showcase

Plug-in Location:

/plugins/synchronize.swf



Function:

Synchronize multiple instances of FSI Viewer.

Syntax:

```
<Plugin src="synchronize" ID="UNIQUE_STRING" />
```

Using the Synchronize plug-in you can synchronize multiple instances of FSI Viewer. To synchronize FSI Viewers you need to use this plug-in in all instances of FSI Viewer you want to synchronize and assign the same unique ID parameter to the instances. The plug-in synchronizes the view (2D and 3D) as well as the mouse cursor and the plug-in measure. The plug-in does not synchronize add-ons like FSI Showcase or FSIPages.

Plug-in Parameters

ID	
Description	Unique identifier
Syntax	String
Default	---
Context	Plug-in attribute

All FSI Viewer instances using this plug-in with the same ID will be synchronized. Please note that this will even be the case if the viewers reside in different browser windows.

SyncSoftwareCursor	
Description	Synchronize SoftwareCursor plug-in
Syntax	Boolean
Default	true
Context	Plug-in attribute
Version	4.1.5 or above

Enable/disable synchronizing the mouse cursor position among synchronized FSI Viewer instances. Note that this parameter has no effect, if you do not use the SoftwareCursor plug-in.

SyncMeasure	
Description	Synchronize the Measure plug-in
Syntax	Boolean
Default	true
Context	Plug-in attribute
Version	4.1.5 or above

Enable/disable synchronizing the Measure plug-in. This parameter has no effect, if you do not use the Measure plug-in. You might want to disable this parameter if you want to enable users to measure distances in synchronized FSI Viewer instances independently.

12.33 ZoomMeter



Plug-in Extends:

- FSI Viewer
- FSI Showcase

Plug-in Location:

/plugins/zoommeter.swf

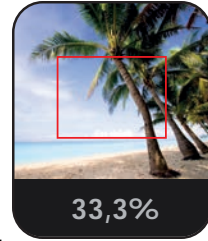
Function:

Displays the current level of magnification.

Syntax:

```
<Plugin src="zoommeter" color="000000" />
```

The "ZoomMeter" plug-in extends the navigator window with a display of the current magnification.



Plug-in Parameters

Color	
Description	Text color
Syntax	HexColor
Default	"000000" (black)
Context	Plug-in attribute

Text color of the zoom display.

The color has to be defined as a 6-digit hexadecimal number. (E.g. "FFFF00" for yellow).

SrcRelative	
Description	Display magnification relative to source image or viewer size
Syntax	Boolean
Default	false
Context	Plug-in attribute
Version	3.0.0 or higher

Display the current magnification in percent of the viewer size (false) or relative to the size of the source image.

13 Appendix

A Escaping or URL-encoding parameter values

In some situations you need to pass escaped (URL-encoded) parameter values to FSI

Viewer, because using unescaped values would break the XML syntax or because you

need to use characters prohibited in URLs.

In cases like this you can pass escaped values rather than passing the actual value.

What does escaping (URL-encoding) mean?

URL-encoding (or percent-encoding) means converting the character to its corresponding value in ASCII and then representing that value as a pair of hexadecimal digits. The digits, preceded by a percent sign ("%"), are then used in place of the reserved character. FSI Viewer transforms URL-encoded values back to the original value.

How can I URL-encode a value?

Most scripting languages (JavaScript, ActionScript, PHP and alike) provide methods for this purpose. Please note that FSI Viewer from version 4 requires Unicode safe encoding rather than non-unicode safe encoding – this is an escaped character

meight be represented by one or more 2-digit numbers preceded by "%".

Using JavaScript you therefore need to use the method "encodeURIComponent()" instead of "escape()".

B HTML Tags available in FSI Viewer

FSI Viewer from version 4 supports basic HTML tags in various contexts like when entering custom tool tips, texts for the "LargeToolTips" plug-in, search results and alike. Whenever this documentation refers to "Basic HTML formatting tags", the following HTML tags can be used (variable values printed **bold italic**):

```
<b></b>
```

```
<i></i>
```

```
<u></u>
```

```
<br />
```

```
<font color="#FF0000" size="12" face="Arial"></font>
```

```
<p align="left|center|right"></p>
```

Please note that you need to use XML syntax in order to avoid XML syntax errors in configuration files - this is e.g. using `
` instead of `
`.

C Example of a `_default.xml` file for FSI Viewer Flash

Example:

```
<fsi_parameter>
  <!-- This file contains default parameters.
  All parameters will be overwritten by additional .xml
  files or query parameters. -->
  <FPX>
    <ServerType value="fsi" />
    <Base value=
      "http://fsi.neptunelabs.com/fsi/
      server?source="/" />
  </FPX>

  <Plugins>
    <Plugins src="mousemodes" />
    <PLUGIN src="zoommeter" SrcRelative="1"/>
  </Plugins>

  <Options>
    <FSIBase value="config/" />
    <ScenePreload value="true" />
    <MenuAlign value="BR" />
  </Options>
</fsi_parameter>
```

D Example of an image specific FSI configuration file

Example:

```

<fsi_parameter>

    <FPX>
        <!-- Src will be prefixed by
        FPXBase of _default.xml -->
        <Src value="3d_object.fpx" />
        <Width value="8128" />
        <Height value="9168" />
        <TilesX value="4" />
        <TilesY value="3" />
    </FPX>

    <Options>
        <InitialAction value="NextScene" />
        <InitialActionDelay value="3" />
        <MenuAlign value="TL" />
        <NoNav value="true" />
    </Options>

</fsi_parameter>

```

E Example of a FSI Viewer JS skin css file**13 Appendix****F Escaping or URL-encoding parameter values**

In some situations you need to pass escaped (URL-encoded) parameters values to FSI

Viewer, because using unescaped values would break the XML syntax or because you

need to use characters prohibited in URLs.

In cases like this you can pass escaped values rather than passing the actual value.

What does escaping (URL-encoding) mean?

URL-encoding (or percent-encoding) means converting the character to its corresponding value in ASCII and then representing that value as a pair of hexadecimal digits. The digits, preceded by a percent sign ("%"), are then used in place of the reserved character. FSI Viewer transforms URL-encoded values back to the original value.

How can I URL-encode a value?

Most scripting languages (JavaScript, ActionScript, PHP and alike) provide methods for this purpose. Please note that FSI Viewer requires Unicode safe encoding rather than non-unicode safe encoding – this is an escaped character might be represented by one or more 2-digit numbers preceded by "%". Using JavaScript you therefore need to use the method "encodeURIComponent()" instead of "escape()".

G HTML Tags available in FSI Viewer

FSI Viewer from version 4 supports basic HTML tags in various contexts like when entering custom tool tips, texts for the "LargeToolTips" plug-in, search results and alike. Whenever this documentation refers to "Basic HTML formatting tags", the following HTML tags can be used (variable values printed **bold italic**):

```
<b></b>
```

```
<i></i>
```

```
<u></u>
```

```
<br/>
```

```
<font color="#FF0000" size="12" face="Arial"></font>
```

```
<p align="left|center|right"></p>
```

Please note that you need to use XML syntax in order to avoid XML syntax errors in configuration files - this is e.g. using `
` instead of `
`.

H Example of a custom skin CSS file

Example:

```

/* IMPORTANT NOTES */
/* - make sure to use a custom class in front of ALL classes
defined here (in this sample: .fsi-skin-example) */
/* otherwise you will break other skins on the same HTML document
*/
/* enables a floating menu bar which overlaps with the image */
/* useful for transparent menu bars*/
/*baseSize defines the size of all skin elements (default 16)*/
/* baseEnlargement defines the general enlargement of the icons
when the viewport is changed, default is 8*/
.fsi-skin-example .fsi-ui-info {
content: '{"floatingMenuBar":true, "floatingNavWindow":false,
"baseEnlargement":10}';
}
/* the main menu bar container */
.fsi-skin-example .fsi-ui-menu-bar {
background: rgba(255, 255, 255, 0.8);
}
/* The logo displayed in the menu bar. Custom logos need to be
placed into the WEBAPPS/fsi/viewer/skins/resources/ folder of
your FSI Viewer installation.
NOTE: Originally, this is defined in the fsi-viewer-core.css.
Since the core css files should not be altered, please overwrite
the classes from there in tyour custom skin CSS. */
.fsi-viewer-root .fsi-skin-example .fsi-ui-logo-image{
background-image:url("[fsi-skin-path]resources/custom_logo.svg");
width: 3.2em;
height: auto;
margin-left:0.15em;
}
/* defines the position of the loading animation */
.fsi-skin-example .fsi-viewer-loading-animation{
left:50%;
top:50%;
transform:translate(-50%, -50%);
}
/* the base button class */
.fsi-skin-example .fsi-ui-button {
background: rgba(255,255,255,0.2);
border:1px solid solid rgba(255,255,255,0.2);
border-radius: 6px;
}
/* styles for hovered buttons */
.fsi-skin-example .fsi-ui-button-hover {
background-color:#FFF;

```

Example:

```
box-shadow: 2px 2px 2px rgba(0,0,0,0.3);
border:1px solid rgba(56,176,170,0.2);
}
/* styles for pressed buttons */
.fsi-skin-example .fsi-ui-button-active {
border:1px solid rgba(56,176,170,0.2);
background:#ffffff;
box-shadow: inset 2px 2px 5px rgba(0,0,0,0.3);
}
/* the base button icon class */
.fsi-skin-example .fsi-ui-button-icon {
color: #33b1aa;
}
.fsi-skin-example .fsi-ui-button-disabled .fsi-ui-button-icon {
color:#bec7c6;
}
.fsi-skin-example .fsi-ui-button-hover .fsi-ui-button-icon {
}
.fsi-skin-example .fsi-ui-button-active .fsi-ui-button-icon {
}
.fsi-skin-example .fsi-ui-menu-bar-disabled .fsi-ui-button-icon{
color: #bec7c6;
}
}
/* select frame */
/* PSEUDO CLASS, only these 2 values will be read */
.fsi-skin-example .fsi-ui-select-frame{
border-color:#33b1aa;
background-color: rgba(255, 255, 255, 0.25);
}
}
/* progress bar displayed when loading image sequences */
/* PSEUDO CLASS, only these 3 values will be read */
.fsi-skin-example .fsi-ui-progressbar{
color:#33b1aa;
background-color:rgba(255,255,255,0.5);
border-color:#999;
}
}
/* PSEUDO CLASS, only the color values will be read */
.fsi-skin-example .fsi-ui-progressbar-text{
color:#555;
}
}
/* tool tips */
.fsi-skin-example .fsi-ui-tool-tip {
color:#4d4d4d;
background-color:rgba(255,255,255,0.8);
border:1px solid #4d4d4d;
```

Example:

```

box-shadow: 2px 2px 2px rgba(0,0,0,0.5);
}
.fsi-skin-example .fsi-ui-tool-tip-arrow {
border-top: 7px solid #4d4d4d;
}
/* navigation window */
.fsi-skin-example .fsi-ui-nav-window {

background-color: rgba(255,255,255,0.8);
}
.fsi-skin-example .fsi-ui-nav-window-image {
border: 1px solid #33b1aa;
background-color: rgba(255,255,255,0.8);
}
/* PSEUDO CLASS, only these 2 values will be read */
.fsi-skin-example .fsi-ui-nav-frame {
border-color: #33b1aa;
background-color: rgba(255,255,255,0.35);
}
/* the text 0.0% to 100.0% below the nav window image */
.fsi-skin-example .fsi-ui-nav-window-zoom-text{
color:#4d4d4d;

}
/* HotSpots */
.fsi-skin-example .fsi-ui-hotspot {
color:#fff;
border:1px solid rgba(255,255,255, 1.0);
border-radius: 2px;
background-color: rgba(88, 168, 255, 0.85);
box-shadow: 2px 2px 2px rgba(0,0,0,0.3);
}
.fsi-skin-example .fsi-ui-hotspot-hover {
background-color: rgba(88, 168, 255, 1.0);
}
.fsi-skin-example .fsi-ui-hotspot-active {
background-color: rgba(88, 168, 255, 1.0);
}
.fsi-skin-example .fsi-ui-hotspot-line-color {
color: #336699;
}
.fsi-skin-example .fsi-ui-hotspot-line-background-color {
color: #113399;
}
/* dialogs */

```


Example:

```
.fsi-skin-example .fsi-dialog{
border:0px none;
background-color:#FFF;
color:#206662;
}
.fsi-skin-example .fsi-dialog input[type="button"]{
font-size: 14px;
color: #666;
border:1px solid #eee;
background-color: transparent;
}
.fsi-skin-example .fsi-dialog input[type="button"]:hover{
color: #206662;
background-color: #EEE;
}
/* FSI Pages */
.fsi-skin-example .fsi-ui-button.PageInput{
background-color: white;
border:1px solid #CCC;
color: #333;
}
.fsi-skin-example .fsi-ui-button.PageInput input:focus{
border-color:#999;
color: #000;
}
```


Index

Appendix	506	imageBorderColor	123
FSI ImageFlow	110	imageServerTemplate	119
Embedding viewer at runtime via		imagesources	113
JavaScript	126	mirrorHeight	120
How it works	111	mirrorStartOpacity	121
JavaScript Interface	128	paddingBottom	118
Callbacks	129	paddingTop	120
callBackClick	130	placeholderColor	123
callBackFocus	130	positionQuantization	115
callBackStart	130	preloadCount	116
callBackZoomEnd	131	presentationType	123
callBackZoomStart	130	quality	120
Methods	128	scrollAlphaActive	117
assignFSIViewer	129	scrollAlphaNormal	117
closeZoom	129	scrollColor	122
destroy	128	scrollHeight	117
focusImage	129	scrollPadding	117
getBuild	129	transparency	114
getInitDone	128	zIndexZoom	126
getInstanceID	129	zoomBackgroundColor	124
getVersion	129	zoomBorderColor	125
init	128	zoomBorderStyle	125
printAPI	128	zoomBorderWidth	124
zoomImage	129	zoomCenteredImageOnly	114
Parameters	111	zoomFadeColor	125
autoDestroy	126	zoomFadeOpacity	125
autoResize	114	FSI Pages Add-on	246
autoRotateSpeed	118	Advanced Parameters	309
backgroundColor	112	Aspect Ratio of the Pages	247
backgroundDarkness	121	Basic Parameters	277
centerZoomedImage	114	Cropping	249
cropRect	119	Defining Image Collections	250
cropValues	118	Event Notifications and Actions	272
curveHeight	122	FSI Pages Converter	247
debug	112	FSI Pages Presets in FSI	
depthFadeout	121	Server Interface "Publish to	
dir	113	Web"	140, 271
duplicateImagesIfNeeded	116	Hyperlinks	258
effects	119	Layout and Skins	248
elementSpacing	115	Link Parameters and Hierarchy of	
elementWidth	115	Link Parameters	265
enableScrollBar	122	Modifying Link Values at Runtime	269
enableZoom	124	Page Overlays	270
endlessScrolling	116	Page Reading Order	252
headers	113	Parameters defining Layout and	287

Appearance	
Parameters for FSI Pages Thumbnails	303
Parameters for Links on Pages ..	297
Parameters for Statistics	307
Parameters for the Front- and Backcover	284
Parameters to retrieve Image Collections	274
Printing Pages	253
Providing XML page data to FSI Pages	259
Publishing PDF Documents with Links using FSI Pages Converter and FSI Server	259
Relative and Absolute Links	267
Saving Pages	254
Saving PDF or other document types	254
Searching in FSI Pages	256
Special URL Values	268
Using FSI Pages	246
FSI Pages JS	134
Aspect Ratio of the Pages	138
Defining Image Collections	136
FSI Pages JS Parameters	140
General	141
Layout/ Appearance	148
Link	160
Plugin	165
How it works	134
JavaScript Interface	170
Callbacks	176
onAfterRegisterExternalButton ..	177
onAutoPageLayoutChange ..	179
onBookmarkAdded ..	179
onBookmarkRemoved ..	179
onBookmarksLoaded ..	179
onBookmarksSaved ..	179
onChangeConfig	176
onDestroy	176
onFullScreen	178
OnMenuButtonEnabled ..	177
onMenuButtononMenuButtonClicked ..	177
OnMenuButtonPressed ..	177
onMenuButtonReleased ..	177
onMenuButtonSetPressed ..	177
OnMenuDestroy	176
onMenuEnabled	176
onMenuReady	178
onOpenPageLink	180
onPageChanged	178
onPageLayoutChange ..	179
onPageSizeChanged ..	180
onPageToolTip	180
onPageZoomed	180
onPrintPage	180
onReady	176
onRegisterExternalMenuButtons ..	177
onRemoveAllBookmarks ..	179
onResize	180
onShowBookmarkList ..	178
onShowChapter	178
onShowPageIndex	178
onShowPrintDialog	178
onSkinChanged	181
Methods	170
addListener	171
changeConfig	170
clickMenuButton	172
destroy	171
firstPage	171
getBuild	174
getConfigValue	175
getCurrentPageNumber ..	173
getInFullScreenMode ..	173
getInitDone	171
getInstanceID	170
getMenuButtonPresent ..	172
getPageContainerDim ..	173
getPageLayoutData ..	175
getPageNumberString ..	173
getParameters	175
getPluginParameter	176
getSkinClassName	173
getTargetPage	174
getVersion	174
goToPage	174
gotoPageNumeric	174
hideMenuToolTip	175
init	170
lastPage	172
makeBoolean	175
nextPage	171

-
- pressMenuButton 172
 - previousPage 171
 - printAPI 170
 - registerExternalMenuButton 172
 - register- 17-
 - ExternalMenuButtonsFromContainer 2
 - releaseMenuButton ... 172
 - removeListener 171
 - setFullScreenElement . 173
 - showAboutWindow 171
 - showMenuToolTip 174
 - start 170
 - togglePageIndex 175
 - unregisterExternalMenuButton 172
 - unre-
 - gister- 17-
 - ExternalMenuButtonsFromContainer 3
 - Parameters
 - adaptiveUISize 143-144
 - autoDoublePageMode 151
 - BackCover 152
 - BackCoverImage 153
 - bendEffect 152
 - BlankBackCover 153
 - ButtonsFirstLastPage 145
 - cfg 143
 - cropRect 149
 - cropValues 149
 - CustomPageNumbers 157
 - CustomPageNumbersFile 158-159
 - debug 141
 - DefaultJavascriptTarget 164
 - DefaultLinkTarget 163
 - DefaultLinkTip 165
 - DefaultLinkURL 163
 - DefaultLinkURLPrefix,
 - DefaultLinkURLSuffix 163
 - DemoURL 146
 - DemoURLTarget 147
 - dir 142
 - doublePageMode 151
 - effects 150
 - emptyImages 1, 154
 - FirstPageNumber 157, 1
 - FollowLinks 161
 - ForceJavascriptTarget 164
 - ForceLinkTarget 164
 - ForceLinkTip 165
 - ForceLinkURL 162
 - ForceLinkURLPrefix 162
 - FrontCover 152
 - FrontCoverImage 153
 - hideUI 143
 - IdleAutoTurn 154
 - IdleAutoTurnDelay 155
 - IdleAutoTurnMaxPage 155
 - IdleAutoTurnMinPage 155
 - Index 145
 - InitialPage 148-149
 - linkRGBAActive 161
 - linkRGBAAHover 160
 - linkRGBANormal 160
 - Links 160
 - LinkTemplateData 162
 - LinkTemplates 161
 - ListTemplate 1, 146
 - MaxZoom 144
 - noNav 141
 - onePageInView 151
 - PageDataTemplate 146
 - PageInput 174, 1
 - PageInputEnterButton 145
 - PageLayout 148
 - PageNumbers 156
 - plugins 165
 - Bookmarks
 - PageNumbers 168
 - PersistentStorage 168
 - PersistentStorageExpiresAfter 168
 - ShowIconsOnPages 167
 - ThumbSize 167
 - Chapters 168
 - FullScreen 165
 - PagesThumbBar 166
 - Autohide 166
 - Height 167
 - indexDataFile 169
 - PageNumbers 166
 - Width 169
 - StickyNotes 167
 - Print 1, 147
 - PrintResolution 1, 147-148
 - RememberLastViewedPage .158

removePages	1, 154	DoublePageZoomScale	283
RomanPageNumbersOffset	157	DropShadow	289
ThumbSize	150	DropShadowDistance	289
UseRomanPageNumbersToPage	156	EmptyImages	275
verticalLayout	150	Events	325
Zoom	144	FirstPageNumber	277
Printing Pages	138	FlipCornerSize	326
Special URL Values	139	FlipEdgeSize	326
FSI Pages mobile	328	FollowLinks	297
How to embed FSI Pages mobile	328	ForceJavascriptTarget	299
Pages Mobile Parameters	329	ForceLinkTarget	299
AutoElevateToHTTPS	331	ForceLinkTip	300
Pages Mobile Plug-in Attribute tar- getdevice	331	ForceLinkURL	299
Parameter		ForceLinkUrlPrefix	300
DeferLinksToOpener	330	ForceLinkUrlSuffix	300
InstructionsImageLandscape	329	ForceSaveURL	317
InstructionsImagePortrait	330	ForceXMLDataFiles	284
MobileDoublePageRatioLimit	329	FrontCover	284
MobileQuitButton	330	FrontCoverConfig	286
Requirements	328	FrontCoverImage	285
FSI Pages Parameter		FullBackCover	285
AutoCropPages	280	FullFrontCover	285
AutoZoom	310	HardPages	286
BackCover	284	HelpImageURL	290
BackCoverConfig	286	IdleAutoTurn	304
BackCoverImage	285	IdleAutoTurnDelay	304
BendEffect	288	IdleAutoTurnMaxPage	304
BendEffectIntensity	288	IdleAutoTurnMinPage	304
BlankBackCover	285	ImageAlign	289
BlankFrontCover	285	Index	278
ButtonsFirstLastPage	306	InitialPage	277
ClickPageToZoom	281	KeepIndex	279
CustomPageNumbers	296	LinkFillAlpha	297
CustomPageNumbersFile	296	LinkRGBAAActive	298
DefaultJavascriptTarget	301	LinkRGBAAHover	298
DefaultLinkTarget	301	LinkRGBANormal	298
DefaultLinkTip	302	LinkTemplateData	303
DefaultLinkURL	301	LinkTemplates	302
DefaultLinkUrlPrefix	302	ListTemplate	275
DefaultLinkUrlSuffix	302	MarginBottom	292
DemoURL	305	MarginLeft	292
DemoURLTarget	305	MarginRight	292
Dir	275	MarginTop	292
DoublePageAllowInteractivity	282	Menu	303
DoublePageZoom	283	MovePages	313
DoublePageZoomDragPan	283	PageBackgroundImage	286
		PageBackgroundImageEffects	287

PageBorderColor	290	SaveUrlSuffix	317
PageCacheLimit	311	SaveURLTarget	317
PageCacheRemoveTime	311	ScrollArrow	291
PageColor	290	ScrollBaseColor	291
PageInput	306	ScrollTrack	291
PageLogDelay	309	Search	320
PageLogEvents	308	SearchAutoExactMatch	322
PageLogTarget	308	SearchAutoSelectFirstResult	321
PageLogURL	307	SearchAutoWildCards	321
PageMarginBottom	292	SearchContext	320
PageMarginInner	292	SearchCustomURL	323
PageMarginOuter	292	SearchCustomValue	323
PageMarginTop	292	SearchDialogHeight	321
PageNumbers	293	SearchDialogWidth	321
PageNumBGColor	295	SearchParameters	323
PageNumColor	294	SearchQueryTemplate	324
PageNumIndent	294	SearchResultClickAction	322
PageNumPos	293	SearchResultServerTemplate	324
PageNumSize	294	SearchSortResults	325
PageOrderRTL	277	SearchTextOnStart	322
PageOverlays	312	SearchType	320
PageOverlayShowZoomed	313	SearchUseMethodGet	325
PageTurnAnimation	279	Server	274
PageTurnSpeed	279	ShowLinksWhileFlipping	297
PreloadBackward	311	ShowLoadProgress	280
PreloadForward	310	SingleSided	287
Print	314	Skin	278
PrintEffects	315	Slider	278
PrintOptions	315	ThumbPadding	293
PrintResolution	315	ThumbSize	293
PrintTemplate	316	UseRomanPageNumbersToPage	295
Ratio	287	ViewerMarginBottom	292
RememberLastViewedPage	309	ViewerMarginLeft	292
Remem-berLastViewedPageExpireAfter	31-0	ViewerMarginRight	292
RemovePages	276	ViewerMarginTop	292
ReversePageOrder	276	XMLBase	313
RomanPageNumbersOffset	295	XMLBaseUseImageFileName	314
Rotation	306	Zoom	280
Save	316	ZoomAreaAnimationSpeedIn	281
SaveAllowSavingImages	318	ZoomAreaAnimationSpeedOut	281
SaveDocumentFile	316	ZoomAreaAnimationType	282
SaveEffects	318	ZoomAreaTargetScale	282
SaveOptions	319	FSI QuickZoom	98
SaveResolution	318	How it works	99
SaveTemplate	319	Initializing FSI QuickZoom manu-ally	105
SaveURLPrefix	317		

JavaScript Interface	106	HoverRotation	354
Callbacks	108	HoverZoom	354
fnPositionAndSizeCallback	108	HScroll	340
Methods	106	ImageBorderHighlight	351
addImages	107	ImageBorderShadow	351
destroy	107	ImageBorderWidth	350
getBuild	107	ImageHeight	349
getInstanceID	107	ImageMarginX	350
getVersion	107	ImageMarginY	350
init	107	ImageWidth	349
printAPI	107	InitialImage	342
removeImages	108	InitialSize	340
scanForNewImages	108	KeepImageSection	360
Parameters	99	LabelContent	355
addCSSClass	101	LabelMarginBottom	358
box-shadow	104	LabelMarginTop	358
debug	99	LabelMode	355
disableByCSSClass	101	LabelOffsetLeft	359
enableByCSSClass	100	LabelOffsetTop	359
fade-in	104	LabelOnRight	359
headers	100	LabelPrefix	357
magnification	101	LabelTemplates	356
maxZoomWindowHeight	102	LabelTextAlign	357
maxZoomWindowWidth	102	LabelTextColor	358
minZoomWindowHeight	103	LabelTextFont	358
pageMargin	104	LabelTextHeight	357
skipImageHeight	103	LabelTextSize	355
skipImageWidth	103	LabelTextWidth	357
useDevicePixelRatio	100	Layout	339
zoomImageBorderWidth	104	MarginBottom	353
zoomImageOffset	104	MarginLeft	353
FSI Showcase		MarginRight	353
Parameter		MarginTop	353
Align	339	MinimizeOnSelect	343
Background	347	OffsetBottom	353
BackgroundAlpha	347	OffsetLeft	353
BaseColor	338	OffsetRight	353
BaseScale	338	OffsetTop	353
CropThumbnails	360	PaddingBottom	354
DragBar	341	PaddingLeft	354
DragBarHighlight	348	PaddingRight	354
DragBarRoundBevel	349	PaddingTop	354
DragBarShadow	349	PreloadThumbs	360
DragBarWidth	341	ScrollArrow	348
DragMenu	341	ScrollBaseColor	347
DragMenuWidth	342	ScrollTrack	348
EdgeRadius	353	ShowControls	344

TextOutsideFrame	343	disableByCSSClass	93
ThumbBorderHighlight	352	enableByCSSClass	93
ThumbBorderShadow	352	loadingImage	92
ThumbBorderWidth	352	monitorPositions	91
ThumbEffects	344	useDevicePixelRatio	91
ThumbFace	351	useTiledImages	92
ThumbspacingX	350	zoomPrecision	92
ThumbspacingY	350	FSI Viewer	182
ToolTips	343	Automated Implementation of	
UseInitialView	342	Images	238
FSI Showcase Add-on	332	Configuration	188
Advanced Parameters	346	Applying Effects to Specific	
Basic Parameters	338	Images Only	218
Defining Image Lists	332	Basic Parameters	195
FSI Showcase Layout	335	Configuration Files	193
ImageList Parameters	344	Default Configuration	193
Showcase Parameters	337	Hierarchy of Configuration	
Using FSI Showcase	332	Parameters	192
FSI Showcase JS	132	Passing Parameters by HTTP	
FSI Showcase Parameter		Query	194
Dir	345	Retrieving Parameters from	
EmptyImages	346	Imaging Servers	194
ListTemplate	346	Using Custom Language	
Server	345	Files	19, 226
FSI TouchZoom	88	Using HTTP Queries	191
Initializing FSI TouchZoom		Using XML Configuration Files	189
manually	89, 93	Debug Mode	236
JavaScript Interface	95	Enabling the Debug Mode	20, 236
Methods	95	Using the Debug Window	237
addImage	95	FSI Viewer 360° Parameter	
destroy	96	Images3D	230
enable	96	Images3DPrefix	231
getBuild	96	Images3DSuffix	231
getVersion	96	Images3DURL	231
init	96	NoSceneAnimation	235
lockImage	97	NoSceneLoop	38, 236
lockImageSourceImage	97	NoSetLoop	38, 235
lockUpdates	97	ScenePreload	234
printAPI	96	SceneSets	233
removeImages	95	TilePaddingX	235
resetImage	97	TilePaddingY	235
scanForNewImages	95	TilesX	232
setZoomPrecision	96	TilesY	232
triggerRefresh	97	Index structure of FSI Viewer	15, 186
updateImagePositions	96	Licensing FSI Viewer	186
Parameters	90	Integration into HTML Pages	187
debug	91	Required HTML-Source Code	187

Introduction	
Additional Buttons	184
Compatibility and User System Requirements	182
Keyboard Shortcuts	185
Mouse Modes	183
The Menu Bar	13, 183
Using FSI Viewer	12, 182
Using the navigator window	185
Parameter	
addConfigFileExtension	225
Animation	212
AnimationSpeed	213
AnimationType	212
AutoElevateToHTTPS	215
AutoHideNav	203
cfg	195
ConcurrentRequests	223
CropRect	201
CropValues	201
debug	199
DefaultCfg	196
DetailBuffer	203
Effects	215
EnableResetInPanMode	209
EnableZoomInPanMode	209
ExtendedViewport	211
FlickMotion	210
FlickMotionOverDrag	211
FSIBase	200
HandCursor	209
height	198
HelpURL	207
HelpURLTarget	208
HideUI	213
IgnoreQuery	220
IgnoreQueryParameters	220
ImagePanButtons	202
InfoTemplate	223
Initial MouseMode	208
InitialAction	206
InitialActionDelay	207
InitialActionPersistent	207
InitialView	205
InitialViewPersistent	206
intro	199
language	202
MagnifySmallImages	212
MaxZoomLevel	221
MenuAlign	201
MetaDataQueryParameters	223
MobileWindowTitle	215
MouseVelocity	225
NavEffects	219
NavFrameColor	204
NavHeight	204
NavMaskAlpha	205
NavMaskColor	204
NavWidth	204
NoImageBlend	214
NoNav	203
path	197
RoundedSkinCorners	213
SceneEffects	219
server	196
serverTemplate	197
skin	198
Src	242
StrictViewport	211
TileEffects	219
TileSizeX	222
TileSizeY	222
TiltModeRelative	222
UISwitchable	214
UniqueID	224
UseIntermediateZoomLevels	224
ViewerBorder	214
width	198
ZoomCache	221
ZoomCacheID	221
ZoomLimit	210
ZoomLimitMin	210
Plug-ins	241
Deactivating Plug-ins	242
Defining Plug-in Parameters	243
Defining Plug-in Parameters dynamically	243
Virtual 360° Presentations	226
Parameters for Virtual 360° Presentations	230
FSI Viewer JS	10
Additional Buttons	14
camera Section	42

Configuration	15	clickMenuButton	78
Hierarchy of Configuration Parameters	17	destroy	75
Retrieving Parameters from FSI Server	19	getBuild	75
Using XML Configuration Files	15	getConfigValue	80
Creating and using Spin Videos	70	getHotspotViewString	77
Embedding viewer at runtime via JavaScript	71	getImageIsTransparent	79
Global Parameters for Hot Spots	39	getInFullScreenMode	79
hotspots3d Section	51	getInitDone	74
How it works	11	getInstanceID	75
JavaScript Interface	73	getMenuButtonPresent	78
Callbacks	81	getMenuHeight	75
onAfterRegisterExternalButton	84	getMouseMode	74
onAnimationComplete	83	getParameters	80
onAnimationStart	83	getPluginParameter	80
onChangeConfig	83	getSkinClassName	79
onChangeImage	83	getSpinFrames	79
onDataComplete	86	getSrcDim	79
onDestroy	82	getVersion	75
onFullScreen	86	getViewString	77
onHotspotEvent	86	getZoom	76
onInitMenu	85	gotoHotspot	77
onMenuButtonClicked	85	gotoViewString	77
onMenuButtonEnabled	84	hideMenuToolTip	80
onMenuButtonPressed	84	init	74
onMenuButtonReleased	85	makeBoolean	80
onMenuButtonSetPressed	85	pressMenuButton	78
onMenuDestroy	83	printAPI	74
onMenuEnabled	83	registerExternalMenuButton	78
onMenuReady	83	releaseMenuButton	78
onMouseModeChanged	85	removeListener	75
onProgress	84	resetView	76
onReady	82	setFullScreenElement	79
onRegisterExternalMenuButtons	84	setMouseMode	75
onReset	85	setZoom	76
onResize	84	showAboutWindow	76
onSkinChanged	84	showMenuButton	76
onStartDragging	85	showMenuToolTip	80
onStopDragging	86	spinToDegree	74
onViewChanged	86	spinToTarget	74
onZoomChanged	83	start	74
Methods	73	updateSize	76
addListener	75	Mouse Modes	13
changeConfig	73	Parameters	20
changeImage	73	actionParameter	58
		activeClass	61
		adaptiveUISize	33
		armAngleMeasureSetup	48

armEndAngle	49	horizontalMarkerPosition	54
armJointPosition	48	horizontalMarkerSize	53
armLength	47	hotspotCallbackFunction	39
armStartAngle	48	hotspotDotColor1	41
autoDestroy	36	hotspotDotColor2	41
autoDisablePointerActions	34	hotspotDotRadius	41
autoHideNav	23	hotspotLineColor1	40
AutoSpin_Button	68-69	hotspotLineColor2	40
AutoSpin_Speed	68	hoverClass	61
autoSpinSpeed	22	imagesources	27
backgroundColor	28	imagesrc	27
base	64	initialMouseMode	36
baseRotationY, baseRotationX, baseRotationZ	50	initialview	31
captureFrames	65	initialViewPersistent	31
captureOffsetEnd	65	inPlaceZoom	33
captureOffsetStart	65	interactive	58
cfg	22	loadExactPreviewImagesAfterResize	35
cssClass	62	markerAngle	55
cssStyle	63	maxZoom	35
cubeLineWidth	49	menuButtonOrder	36
cubeSizeX	49	MouseModes_MenuOffset	69
cubeSizeY	49	MouseModes_Mode n	69
cubeSizeZ	49	MouseModes_Sequence	70
debug	21	mouseSensitivity	28
decodeHTMLEntitiesInHotSpotTexts	41	navHeight	23
desktopClickZoomScale	26	navWidth	23
digitalCropFactor	42	noNav	23
dir	27	normalClass	61
distanceToTableCenter	45	panLimits	32
documentZoom	34	pellets	28
drawCube	29	pelletsOpacity	29
drawGroundPlate	30	perimeterAngle	55
drawTable	29	perimeterAngleFadeOut	56
effects	30	perimeterAngleFadeOutX	57
enableHotspots	39	perimeterAngleRange	56
enableZoom	26	perimeterAngleRangeX	56
focallength	43-44	perimeterAngleX	55
focusAngleX, focusAngleY	60	perimeterImageWidth	57
focusOnClick	59	perimeterZoom	57
focusRadius	60	plugins	67
frameOffset	67	Plugins	67
headers	21	position	51
heightCamera	45	preventImageTransparency	33
heightCorrectionFactor	66	progressBarHAlign	24-25
heightTable	45-46	progressBarHeight	24
hideUI	22	progressBarWidth	24
		reverseSceneSets	38

rotateOnFocus	59	Plug-in Reference	362
rotationXRange, rota-		AddThis	363
tionYRange	50	Parameter	
sceneSet	64	AppendFSIParameters	367
SceneSets	37	BookmarkTitle	365
shiftX	50	BookmarkURL	365
shiftY	50	DialogHeight	364
shiftZ	50	DialogWidth	364
showPreviewWhileLoading	31	Library	367
showProgressbar	32	ListServices	366
skin	25	PriorityServices	366
source	63	RemoveServices	366
spinDirection	64	Services	365
spinDirectionImage	37	ShareURLTemplate	367
spinPreviewWhileLoading	32	ShowLabels	364
tableDiameter	30	ShowTooltips	364
targetHeightOffset	46	TargetFrame	363
targetWidthOffset	46	UserName	363
text	53	BackgroundImage	369
url	58	Parameter	
urlLoadingAnimation	25	Effects	370
urlTarget	59	FPXSrc	369
useDevicePixelRatio	33	Chapters	371
verticalMarkerPosition	54	Parameter	
verticalMarkerSize	53	ColoredIndex	373
widthCorrectionFactor	66	ComboLabelBackgroundAlpha	373
x	52	IndexData	372
xCorrectionFactor	66	IndexDataFile	372
y	52	IndexMarginWidth	373
yCorrectionFactor	67	PageNumbers	373
z	52	Sort	374
zoomOnFocus	60	Update	374
Using the navigator window	15	ValidateData	375
video Section	63	Width	374
virtualSpace Section	49	ClockProgress	376
Interface		Parameter	
Methods		Alpha	376
register-	7-	Color1	377
ExternalMenuButtonsFromContainer	8	Color2	377
unregisterExternalMenuButton	78	LineColor	378
unre-		posX	378
gister-	7-	posY	379
ExternalMenuButtonsFromContainer	9	Size	377
Introduction	8	TextColor	378
Parameter	92	ColorAdjust	380
		Parameter	
		AutoHideDialog	380

Blue	380	Tip	401
Brightness	381	URL	403
Green	380	View	401
MenuOffset	381	Visible	395
Red	380	Imprint	408
SetValuesForEachImage	381	Parameter	
ShowUI	382	AddButton	409
ContextMenu	383	DataFile	408
Parameter		DialogHeight	410
MenuItems	384	DialogWidth	410
CustomButton	385	MenuOffset	409
Parameter		JavaScript Bridge	411
[ID].Frame	386	Parameter	
[ID].LabelFrame	387	AllowDomains	422
[ID].Offset	386	Callback	423
[ID].ToolTip	387	Explore	422
[ID].URL	386	ImageUrls	423
Buttons	385	LargeToolTips	424
MenuOffset	385	Parameter	
Fullscreen	390	BackgroundAlpha	426
History	392	BackgroundColor	426
Parameter		CornerRadius	428
Length	392	Delay	425
Loop	393	HighlightColor	426
MenuOffset	392	LineWidth	427
SmallButtons	393	Shadow	425
HotSpots	394	ShadowColor	426
Parameter		TextColor	427
Alpha	396	TextFont	427
BaseURL	396	TextSize	428
CFG	404	Width	425
Color	403	Magnifier	429
Coords	401	Parameter	
DefaultColor	398	Handle	431
DefaultFill	398	HandleSize	431
DefaultFillAlpha	399	Magnification	430
DefaultFillColor	399	MenuOffset	429
DefaultMove	397	Reflections	431
DefaultSkew	398	Size	430
DefaultTarget	397	Visible	430
Events	396	MaxZoom	432
Fill	403	Parameter	
Move	402	MenuOffset	432
SelectImage	405	Measure	433
Skew	402	Parameter	
Spot	400	BGAlpha	439
Target	404	BGColor	438

Decimals	436
Events	440
GlowAlpha	439
GlowColor	440
GlowRange	439
ImageWidth	435
LineColor	437
MenuOffset	435
Prefix	436
ShowAngle	437
ShowText	436
Suffix	436
TextBorder	438
TextColor	437
TextSize	438
Mousemodes	442
Parameter	
MenuOffset	442
Moden	442
Sequence	443
MousemodeSelect	444
Parameter	
MenuOffset	444
Moden	444
Music	446
Parameter	
AutoPlay	446
KeepState	448
Loop	447
MenuOffset	446
PauseOnStop	447
Song	447
Volume	448
NavExtension	449
Parameter	
MenuOffset	449
Visible	449
X	450
Y	450
Notepad	451
Parameter	
AddButton	458
BackgroundColor	454
BackgroundGradientAngle	455
EdgeRadius	456
EnableAddCustomBookmark	457
EnableAddFullBookmark	457
EnableAddLinkBookmark	457
EnableSendBookmarks	457
ExpandSelectionToImage	458
FlipAndZoomInOneGo	459
FrameColor	455
FrameWidth	455
Height	454
ImageBackgroundColor	456
ImageEffects	458
ImageHeight	454
ImageWidth	454
ItemButtonsBackgroundColor	456
LoadXML	460
LoadXMLPrefix	461
MenuOffset	459
Revision	453
ScrollArrow	457
ScrollBaseColor	457
ScrollTrack	457
SendXMLStoreURL	461
SendXMLUrlFailed	462
SendXMLUrlFailedTarget	462
SendXMLUrlSuccess	461
SendXMLUrlSuccessTarget	462
TextColor	457
UniqueID	453
Visible	453
Width	454
PagesLibrary	463
Parameter	
AddButton	464
DataFile	463
DialogHeight	466
DialogWidth	466
ImageHeight	464
ImageWidth	464
MarginX	465
MarginY	465
MenuOffset	465
ShowLabels	465
PagesMirror	467
Parameter	
Blur	467
EndAlpha	468
Offset	467
StartAlpha	468

PageSounds	469	SelectFrame	484
Parameter		Parameter	
Library	469	AnimatedRect	485
Volume	469	AspectRatio	486
PagesThumbBar	470	Callback	487
Parameter		Events	488
AlignLeft	470	InvertShift	487
ArrowAlphaActive	471	MaskAlpha	486
ArrowAlphaNormal	471	MaskColor	486
ArrowFillColor	471	MenuOffset	484
ArrowLineColor	472	Rotate	485
Arrows	470	Visible	485
DropShadow	472	Window	485
Height	472	ShoppingList	489
HideOnZoom	473	Parameter	
MagnifyCurrentPages	473	Currency	489
MarginBottom	473	DecimalCharacter	490
MarginTop	474	DefaultUnit	490
OffsetLeft	474	DialogHeight	493
OffsetRight	474	DialogWidth	494
Padding	475	Email	490
SelectionBorderWidth	475	MaxCountPerItem	491
SelectionColor	475	MinItemHeight	491
SelectionMarginBottom	476	PersistentStorage	491
SelectionRoundCorner	476	PersistentStorageExpiresAfter	492
TextColor	476	Print	492
TextColorHighlight	477	ShowListOnAddItem	492
PrintSave	478	TargetURL	493
Parameter		Template	493
MenuOffset	478	SoftwareCursor	495
Print	478	Parameter	
PrintEffects	479	Library	495
PrintResolution	479	StickyNotes	496
PrintTemplate	479	Parameter	
Save	480	AllowMirroredNotes	496
SaveEffects	480	Alpha	496
SaveResolution	480	Color	497
SaveTemplate	481	ColorHighlight	497
Resize	482	ColorShadow	498
Parameter		CornerRadius	500
EnlargeBy	483	NaturalRotation	498
MaxHeight	482	PageNumbers	498
MaxWidth	482	PersistentStorage	499
MinHeight	483	PersistentStorageExpiresAfter	499
MinWidth	483	Position	499
		Scale	500
		Spacing	500

TextColor	497
Synchronize	501
Parameter	
ID	501
SyncMeasure	502
SyncSoftwareCursor	502
ZoomMeter	503
Parameter	
Color	503
SrcRelative	503
useTiledImages	92

FSI Viewer

NeptuneLabs GmbH

Lagesche Str. 32
32657 Lemgo
Germany

Fon: +49 5261-28732-0

Fax: +49 5261-28732-29

eMail: info@neptunelabs.com

http: www.neptunelabs.com

www.fsi-server.com

No part of this manual, including the software described in it, may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form or by any means, except documentation kept by the purchaser for backup purposes, without the express written permission of NeptuneLabs.

Specifications and information contained in this manual are furnished for informational use only and are subject to change at any time without notice, and should not be construed as a commitment by NeptuneLabs. NeptuneLabs assumes no responsibility or liability for any errors or inaccuracies in this manual, including the software described in it.

© 2009-2017 NeptuneLabs GmbH, Germany. All rights reserved.

www.fsi-viewer.com

Neptune
L a b s

Internet Application Development

NeptuneLabs GmbH

Lagesche Str. 32

32657 Lemgo

Germany

fon: + 49 5261 - 28732- 0

fax: + 49 5261 - 28732- 29

info@neptunelabs.com

www.neptunelabs.com