ISBN n/a

Frogans Slide Description Language - 3.0 - Extracts

<resimage>, <respixels>, <resdraw>, <respath>, and <layer> elements

Abstract

This document sets forth the third generation of the markup language used to create Frogans slides. A Frogans slide is an interactive and highly visual transparency which is part of a Frogans site. A Frogans slide is rendered identically by Frogans Player on all end-user devices regardless of the screen size. A Frogans slide is either static or generated dynamically by the server hosting the Frogans site. This language, based on XML, covers both the format used for FSDL documents, each one describing a Frogans slide, and the format used for FSDL-Request documents, each one being sent conditionally by Frogans Player to the server when the end user navigates to a new Frogans slide.

Status

This document is a preliminary draft, for examination, review, and evaluation by the developer community.

Aspects of this document may be discussed publicly on the Frogans technology mailing lists, accessible at the following permanent URL: https://lists.frogans.org/.

Location

This document is accessible at the following permanent URL: https://www.frogans.org/en/resources/fsdl/access.html

Copyright Statement

Table of Contents

1. Introduction

1.1. Background

Started in 1999, the Frogans project aims to introduce a new software layer on the Internet alongside other existing layers such as E-mail or the Web. The goal of this new software layer, called the Frogans layer, is to enable the publishing of Frogans sites.

The Frogans technology developed for the Frogans project is the foundation of the Frogans layer. It includes a markup language used to create Frogans slides, which are the transparencies (or pages) of a Frogans site. This language is called the Frogans Slide Description Language, or FSDL.

Frogans slides are interactive and highly visual. A Frogans slide is rendered identically on all end-user devices regardless of the screen size and regardless of the means for interactivity: pointing device, touch screen, etc. End-user devices today include smartphones, tablets, desktop computers, watches, and virtual reality headsets.

A Frogans site is made up of an unlimited number of Frogans slides hyperlinked to each other. It can be hosted on a server located anywhere in the world. The server is chosen by the publisher of the Frogans site, who can be any individual or organization in any country. End users use Frogans Player to access a Frogans site via its Frogans address, and then navigate the Frogans site from Frogans slide to Frogans slide.

A Frogans slide can be either static or dynamic, i.e. generated on the fly by the server in conjunction with any back-end server such as a database server, in order to automatically update or personalize the Frogans site for the end user.

At the time the FSDL language was originally designed, the goals were to define a language with the following fundamental characteristics:

* It had to be a description language and not a programming language; notably, it had to make it possible to define all end-user interactivity of a Frogans slide using simple descriptors, without having to write scripts.

* It had to enable the creation of Frogans slides in any shape, that appear on screen via non-rectangular windows.

* It had to support all the graphical features of Frogans slides, including image manipulations, color gradients, fonts, right-to-left and vertical text, resource filtering, masks, shadows,

buttons, and the leapout effect.

* It had to be secure by design in order to ensure the protection of end users; for example, Frogans slides could not incorporate programs to be executed on the end-user device.

* It had to be quick and simple to learn and use, even when creating advanced Frogans slides; to that end, it had to integrate into a single language, with a limited number of elements, all aspects of Frogans slides, such as content, layout, style, user interaction, and state information management.

* It had to be based on markups and provide a clear and comprehensible structure closely related to the layout of the Frogans slide, in order to be easily hand-coded, using a basic text editor.

Furthermore, the language had to include strict parsing rules, for three reasons. First, when combined with a helpful error reporting system, this would make it easy for authors and authoring tools to create, maintain, share, and reuse Frogans slides. Second, this would help ensure precise and uniform rendering of Frogans slides, by eliminating the need for implementations such as Frogans Player to guess the author's intentions or recover from errors when parsing non-compliant documents. Third, given that strict parsing rules are more straightforward to implement than loose ones, this would reduce the risk of possible implementation bugs which could lead to security breaches.

The HTML language [HTML] used for structuring and presenting content on the World Wide Web was not chosen as a basis for the FSDL language as it could not directly achieve all of these goals without modifying and drastically simplifying its syntax, while integrating into the HTML language features provided by CSS [CSS] and JavaScript [JavaScript].

The Extensible Markup Language (XML) defined by the W3C [XML] was chosen as the basis for the FSDL language because it presents several advantages: XML offers a foundation for designing languages with strict parsing rules; XML supports internationalization as it is based on the Unicode Standard [Unicode]; and XML-based documents are easily authored, stored, processed, and transmitted. The main disadvantage of XML is its verbosity which makes XML-based documents longer to parse than more concise or binary formats. However, when compared to the processing required to render Frogans slides, and given the relatively small size of FSDL documents, this disadvantage is negligible. XML has since become a mature and widely-adopted standard.

The first generation of the FSDL language achieving the preceding goals was described in informal specifications. The second generation was formally described in versions 2.0 and 2.1 of the Frogans Slide Description Language specification released in 2002 and in 2004 [FSDL-HIST]. All these versions of the FSDL language are no longer in use.

1.2. Purpose

The purpose of this document is to set forth the third generation of the FSDL language.

This new generation must retain all the fundamental characteristics of the FSDL language, while redesigning certain parts, improving capabilities, and adding new features in order to meet the needs of stakeholders involved in the publication of Frogans sites.

The following domains must be addressed: the communications between Frogans Player and servers hosting Frogans sites, the design of Frogans slides, and the protection of end-user interests.

As concerns the communications between Frogans Player and servers hosting Frogans sites:

* The network protocols that could be used for publishing Frogans sites were intrinsically tied to the FSDL specification, meaning that a new version of the specification was required whenever there was a need to change the usage profile of an existing network protocol, or to support a new network protocol.

  The FSDL language must be redesigned to run on top of Uniform Content Server Request (UCSR) [UCSR], a new framework developed by the OP3FT to make the Frogans technology independent from the underlying network communications.

* The method that authors of FSDL documents had to use in order to refer to other FSDL documents and auxiliary files in the Frogans site root directory was complex and offered insufficient functionality. For example, the static or dynamic nature of the referred file was not defined explicitly but depended on the context, and there were no means for sharing an auxiliary file among several resources or for embedding an auxiliary file in an FSDL document.

  The FSDL language must introduce, within an FSDL document, an improved method for referring to other files.

* The method that authors of FSDL documents had to use in order to define return-to-server data associated with referred files (either FSDL documents or auxiliary files) or with Frogans slide sessions did not directly support Unicode characters [Unicode] and offered insufficient functionality. For example, authors of FSDL documents could not share return-to-server data among several referred files and they had to concatenate return-to-server data items into a single value.

  The FSDL language must introduce, within an FSDL document, an improved method for defining return-to-server data.

* When a request containing return-to-server data was sent by Frogans Player to the server hosting the Frogans site, the data groups making up the return-to-server data in the request were not insulated from one another, which made it difficult to add new kinds of groups in the future. In addition, the request did not contain by design any information on whether it was for an FSDL document or for an auxiliary file. Likewise, a request for an FSDL document did not contain any contextual information, such as that the end user triggered a button leading to another Frogans slide of the Frogans site, or that the end user took no action before the next Frogans slide began to load.

  The FSDL language must introduce a new structured document, called the FSDL-Request document, to be sent conditionally by Frogans Player to the server hosting the Frogans site.

* The methods available to Frogans site publishers for allowing end users to enter information to be sent to the server hosting the Frogans site were insufficient and must be replaced. Either the publisher had to set the Frogans address of the Frogans site so as to require the end user to enter a personal identifier or a login and password when opening the Frogans site (this information was sent to the server in the request for the home slide). Or the publisher had to require the end user to enter the information piece by piece through a series of Frogans slides (for example, each Frogans slide would allow the end user to trigger one of several buttons, each button representing a single number, letter, or word).

  The FSDL language must introduce, within an FSDL document, a simple method for allowing end users to enter information, where the entry takes place after the end user has triggered a button in order to navigate to another Frogans slide.

* The method available to Frogans site publishers for allowing end users to download a file of any type from the server hosting the

Frogans site to the end-user device, which was introduced in the second generation of the FSDL language, was potentially dangerous and must be removed. For example, this method could be used by malicious publishers to deliver files containing viruses which could infect end-user devices lacking adequate anti-virus protection.

* The method available to Frogans site publishers for allowing end users to download an image file to be displayed in a larger size on the screen, which was introduced in the second generation of the FSDL language, must be removed. This method was inappropriate for end-user devices with small screens and therefore could drive Frogans site publishers to develop two different versions of the same Frogans site: one for end-user devices with a large screen, and another for those with a small screen.

  Driving Frogans site publishers to develop various versions of the same Frogans site in order to take into account specific characteristics (such as screen size, memory, or processing power) of certain end-user devices obviously runs against the founding principles of the Frogans technology defined in the OP3FT Bylaws [BYLAWS].

As concerns the design of Frogans slides:

* The method for rendering Frogans slides in Frogans Player used an area whose dimensions in pixels depended on screen characteristics of the end-user device (i.e. the size of the screen, its resolution, the resulting pixel density) and the intended distance between the end user and the screen. This could cause the text in a given Frogans slide to be rendered differently on different end-user devices. Indeed, because of the rules required by many writing systems for positioning and substituting glyphs when rendering text, proportionally changing both the dimensions in pixels of a text area and the font size does not necessarily result in an identical appearance for the text, even when using scalable fonts. For example, the number of lines in the text area can change.

  As a result, this method did not fully comply with the founding principle of the Frogans technology whereby a Frogans slide is rendered identically on all end-user devices and must be replaced.

  The FSDL language must introduce a method for rendering Frogans slides using an area whose dimensions in pixels are fixed. In order to take into account screen characteristics of the end-user device and the intended distance between the end user and the screen, the FSDL language must also introduce a new concept,

called the scaling factor, as well as a complementary method based on that concept to be used in Frogans Player for displaying rendered Frogans slides.

* Given the new method for rendering Frogans slides, the virtual grid made up of points representing abstract positions which was used by authors to lay out Frogans slides is no longer relevant.

  The FSDL language must be redesigned so that the layout of Frogans slides is based on a virtual grid made up of pixels, called the Frogans slide rendering canvas.

* Authors of FSDL documents had to define the overall shape of the Frogans slide separately from the content that would fill in the shape of the Frogans slide. Given that the shape was rendered as a binary mask, each pixel of the rendered Frogans slide was either fully opaque or fully transparent. This look has become outdated and must be replaced. End users today are accustomed to seeing windows with several levels of transparency on their screens.

  The FSDL language must be redesigned so that authors of FSDL documents can create Frogans slides directly using superimposed layers, each one with an alpha channel. As a result, Frogans slides no longer necessarily have sharp edges and the FSDL language must therefore also introduce a new method enabling end users to resize Frogans slides on devices with a pointing device, in order to replace the resize method which used the edges of the Frogans slide.

* Authors of FSDL documents already had the possibility of including, in an FSDL document, text to be displayed based on writing systems from around the world (including right-to-left and vertical writing systems). But there were two issues: first, the method for rendering text did not support the rendering of text based on complex scripts (such as Arabic, Devanagari, and Thai); and second, the rendering of text could lead to different results on different end-user devices because the method used by authors of FSDL documents to refer to fonts was based on fonts installed locally on the end-user device, and when a referred font was not in fact installed, then it was substituted by another installed font.

  As a result, the method used by authors of FSDL documents to refer to fonts did not fully comply with the founding principle of the Frogans technology whereby a Frogans slide is rendered identically on all end-user devices and must be replaced.

  The FSDL language must introduce an improved method for rendering

text based on complex scripts. The FSDL language must also introduce, within an FSDL document, a new method for referring to fonts which is exclusively based on a set of selected fonts to be embedded in implementations such as Frogans Player.

As concerns the protection of end-user interests:

* In order to protect the interests of end users, the FSDL language had a limitation on the total size, in bytes, of a Frogans slide including the FSDL document and its auxiliary files. This limitation is important for end users permanently or temporarily connected to a low-bandwidth network, for instance, as well as for end users who subscribe to mobile Internet access services with a monthly data allotment and who, if they exceed their allotment, suffer bandwidth limitations or must pay additional fees.

  However, given the new method for rendering Frogans slides, this limitation on the total size, in bytes, of the Frogans slide could be exceeded in many cases and must be adapted. Indeed, with this new method, the dimensions in pixels of pre-authored images contained in auxiliary files need to be large enough to remain independent from the screen characteristics of the end-user device.

  The FSDL language must continue to limit the size, in bytes, of a Frogans slide, but must include two separate limits: one concerning the maximum size of an FSDL document, and another one, which must be increased, concerning the total size of a Frogans slide including the FSDL document and its auxiliary files.

* When rendering Frogans slides, there was no limitation on the amount of memory required. In particular, the surface, in pixels, of Frogans slide components was not taken into account. The only factor that could limit memory usage was the limitation on the number of elements in the FSDL document corresponding to Frogans slide components, which was applied when the FSDL document was validated. This lack of an effective limitation must be corrected because it could prevent end users having devices with limited memory from browsing certain Frogans sites (whether these devices are older, inexpensive, or designed for a specific use), which obviously runs against the founding principles of the Frogans technology defined in the OP3FT Bylaws.

  The FSDL language must introduce new rules to limit the amount of memory required to render a Frogans slide. These rules must be applied after the FSDL document is validated and before rendering the Frogans slide, and must take into account relevant Frogans slide components defined in the FSDL document.

* When rendering Frogans slides, there was no limitation on the total surface, in number of pixels, of the pre-authored images used. This lack of a limitation also raises memory usage issues and must be corrected.

  The FSDL language must introduce a new rule to limit the total surface, in number of pixels, of all the pre-authored images used to render a Frogans slide. This rule must be applied when rendering the Frogans slide.

* Given the new possibility for authors of FSDL documents to create Frogans slides directly using superimposed layers, each one with an alpha channel, the rules ensuring that end users can see and easily move the Frogans slide on the screen are no longer sufficient. Due to the fact that parts of a Frogans slide can be very transparent, end users could have problems seeing the Frogans slide on the screen, seeing whether any one of the buttons of the Frogans slide has been selected, moving the Frogans slide on the screen, or selecting each button of the Frogans slide.

  The FSDL language must introduce a new set of rules, called usability rules, ensuring that end users can see and easily use the Frogans slide on all devices, regardless of the mode of interactivity: pointing device, touch screen, etc. These rules must be applied when rendering the Frogans slide.

* The method available to Frogans site publishers for enabling a third party to collect statistics about end-user browsing on their Frogans site must be removed because it raised privacy issues. Although this method provided a useful means for independently measuring the traffic on a Frogans site, the third party collecting the statistics could use this method to track end users and compile long-term records of their browsing histories, without informing them, asking their permission, or giving them control over how their personal data was used.

  Given that no solution could be found that enables a third party to collect statistics while fully protecting end-user privacy, this method must be simply removed from the FSDL language and not replaced. The respect of privacy is a founding principle of the Frogans technology.

Furthermore, the third generation of the FSDL language must introduce, in response to requests from authors of FSDL documents, new practical features for designing Frogans slides, such as:

* enhancement of image resources, in order to add support for an image file containing a collection of images, and for embedding an

image file as text characters in the FSDL document

* path resources, intended to provide a means for including Bezier curves in a Frogans slide

* script-adaptive fonts, intended to provide a means for defining fonts to be selected automatically when rendering sequences of characters of a text resource that correspond to different writing systems

* merge resources, intended to provide a means for preparing a new resource that gathers or mixes several other resources

After all the preceding changes have been made to the FSDL language, any future modifications to the FSDL language will be set forth in a new document bearing a new version number. Indeed, as stated in the OP3FT Bylaws, the technical specifications of the Frogans technology are not modified once published.

New versions of the FSDL specification will be prepared as needed, notably in order to take into account feedback and suggestions from the various stakeholders involved in the publication of Frogans sites. For the list of future enhancements under discussion at the time this version of the FSDL specification is being completed, see Section 11).

In order to make it possible for all implementations such as Frogans Player to render Frogans slides identically on all end-user devices, right down to the pixel level, this specification must define the rendering algorithms to be used for resources, for transformations applied to resources, for combining parts of merge resources, for combining layers, and for adding shadows.

1.3. Intended Audience

This document is intended for those involved in the publication of Frogans sites, such as Frogans site publishers, Frogans site designers, Frogans site developers, Frogans site hosts, and Frogans site indexers.

This document is also intended for:

* creators of resources based on the FSDL language, such as online tutorials, videos, blogs, or books

* developers wishing to implement software based on the FSDL language, such as an authoring tool or a CMS (Content Management System)

This document is also intended for anyone wishing to understand the principles and concepts related to the communications between Frogans Player and servers hosting Frogans sites, the design of Frogans slides, and the protection of end-user interests.

7.2. Preparing Image Resources - <resimage> Element

The <resimage> element prepares an image resource that is based on an image file. The image file contains a pre-authored image, which is created previously using image editing software.

The <resimage> element is a child of the <frogans-fsdl> element (see Section 6.1). The <resimage> element does not contain any children. It does not contain text characters.

As stated in the Frogans slide design concepts (Section 4):

- The image resource is intended to be used in the Frogans slide as the foundation of a layer (see the <layer> element in Section 7.12) or as part of a merge resource (see the <resmerge> and <merge> elements in Section 7.11).

- The image resource can be used once or several times in the Frogans slide.

The image resource is identified using a component identifier (Section 6) which is determined by the value of the "resid" attribute of the <resimage> element.

The size, in pixels, of the image resource depends on the value of the "size" attribute of the <resimage> element.

The image file on which the image resource is based is defined using a <file> element (see Section 7.1). As a result, the image file is either a static or a dynamic auxiliary file in the Frogans site root directory (Section 3), or is embedded as text characters in the <file> element, depending on the value of the "nature" attribute of the <file> element.

The image file is referred to using a component identifier which is determined by the value of the "fileref" attribute of the <resimage> element.

The image file is encoded using one of the following image formats: PNG [PNG], JPEG [JPEG], or GIF [GIF].

If the image file contains information on the degree of opacity of its pixels, such as an alpha channel that can be defined using the PNG image format, then this information is used when preparing the image resource. Otherwise, the pixels in the image file are considered fully opaque when preparing the image resource.

As stated in the Frogans slide design concepts, in order to avoid on-

screen distractions (Section 4), if the image file is encoded using the GIF image format and contains an animation, then only the first frame is used when preparing the image resource. Likewise, if the image file is encoded using the APNG image format and therefore contains an animation [APNG], then only the first frame is used when preparing the image resource.

For more information on the process to decode image files, see Appendix B.1.

If the image file is an auxiliary file, then according to Appendix B.1 on decoding image files, the image file extension is not required in the "name" attribute of the <file> element.

As stated in the memory usage rules for protecting end-user interests (Section 5.2), the width and height of the pre-authored image are both limited to 1,024 pixels.

A selection of the pre-authored image is used, representing either the entire pre-authored image, or an extract of the pre-authored image. Using an extract of the pre-authored image can be helpful if, in order to reduce latency for loading the Frogans slide, the pre-authored image is an image sprite, i.e. it contains a collection of images.

The image selection depends on the values of the "selection" and "bounds" attributes of the <resimage> element. By default, the image selection is the entire pre-authored image.

The way in which the image selection is incorporated in the image resource depends on the values of the "aspect", "adjust", and "origin" attributes of the <resimage> element. The size, in pixels, of the image selection is not necessarily the same as the size, in pixels, of the image resource.

If the image file is an auxiliary file, then its size, in bytes, is subject to the overall file size limitations applicable to the Frogans slide stated in the network usage rules for protecting end-user interests (Section 5.1).

If the image file is embedded as text characters in the <file> element, then the text characters represent the contents of the image file encoded using Base64 (see Section 7.1).

Embedding an image file as text characters in the <file> element reduces latency for loading the Frogans slide, because both the FSDL document and the image file are loaded at the same time. Embedding an image file can also simplify the task for generating both the FSDL

document and the image file on the fly (i.e. dynamically) on the server hosting the Frogans site.

However, embedding an image file as text characters in the <file> element leads to an overhead to be taken into account in the overall file size limitations applicable to the Frogans slide (Section 5.1), because the number of text characters required to encode the image file is approximately 1.33 times the size in bytes of the image file. Furthermore, depending on the character encoding of the FSDL document (Section 6), more than one byte may be necessary to encode each text character. Finally, an embedded image file cannot benefit from potential inclusion in the auxiliary file cache (Section 3).

The <resimage> element has the following attributes:

resid - Mandatory

The identifier of the image resource.

The value of the attribute is a component identifier. The value is compliant with the rules presented in Section 6.

As a result, the value cannot have been utilized previously in the FSDL document as a component identifier.

Examples of accepted values:
resid='foo1'
resid='Foo1'
resid='foo2'
resid='foo_1'

Examples of refused values:
resid=''
resid='foo-1'
resid='foo/1'
resid='foo.1'
resid='foo 1'
resid='foo1,foo2'

size - Mandatory

The size of the image resource.

The value of the attribute represents the width and height, in pixels, of the image resource.

As stated in the Frogans slide design concepts (Section 4), the width of the image resource is an integer between 1 and 640

(inclusive) and its height is an integer between 1 and 480 (inclusive).

The value of the attribute is a string in the 'width,height' pattern where: ',' is the U+002C COMMA character; and 'width' and 'height' are substrings representing respectively the width and height of the image resource. Each substring contains between one and three decimal characters (inclusive). Each decimal character of the substring is in the range from '0' to '9' (U+0033 DIGIT ZERO to U+0039 DIGIT NINE, inclusive). The substring does not contain any leading '0' (U+0033 DIGIT ZERO) characters.

As a result, the value of the attribute does not contain any of the following characters: white space characters defined in Section 6, '+' (U+002B PLUS SIGN), '-' (U+002D HYPHEN-MINUS), and '.' (U+002E FULL STOP).

Examples of accepted values:
size='40,200'
size='400,200'

Examples of refused values:
size=''
size='400,0'
size='400,550'
size='400'
size='400 200'
size='040,200'
size='400, 200'
size='+400,200'
size='400,-200'
size='400.5,200'

fileref - Mandatory

The identifier referencing the image file on which the image resource is based.

The value of the attribute is a component identifier. The value is compliant with the rules presented in Section 6.

The value of the attribute has been utilized previously in the FSDL document as a component identifier for a <file> element (see Section 7.1).

Examples of accepted values:

```
        fileref='foo1'
        fileref='Foo1'
        fileref='foo2'
        fileref='foo_1'

        Examples of refused values:
        fileref=''
        fileref='foo-1'
        fileref='foo/1'
        fileref='foo.1'
        fileref='foo 1'
        fileref='foo1,foo2'
```

selection - Optional

The image selection used in the image resource.

The attribute is based on a string set which represents the possible values of the attribute.

The value of the attribute is case-sensitive.

The value can be:

* 'entire' indicates that the image selection is the entire pre-authored image.

* 'extract' indicates that the image selection is an extract of the pre-authored image. The extract depends on the value of the "bounds" attribute of the <resimage> element.

The default value is 'entire'.

bounds - Applicable only if the value of the "selection" attribute equals 'extract'. Mandatory if applicable.

The bounds of the image selection used in the image resource.

The value of the attribute represents the left, top, right, and bottom coordinates, in pixels, of the image selection in the pre-authored image.

The left and top coordinates are integers between 0 and 1,023 (inclusive). The right and bottom coordinates are integers between 1 and 1,024 (inclusive).

The image selection uses an x,y coordinate system where: the x and the y coordinates of the pixel at the top-left corner of the pre-

authored image are both zero; values on the x axis increase towards the right; values on the y axis increase towards the bottom.

The image selection is made as follows:

* The left coordinate is equal to the x coordinate of the pixel at the top-left corner of the image selection.

* The top coordinate is equal to the y coordinate of the pixel at the top-left corner of the image selection.

* The right coordinate is equal to one plus the x coordinate of the pixel at the bottom-right corner of the image selection. The right coordinate is greater than the left coordinate.

* The bottom coordinate is equal to one plus the y coordinate of the pixel at the bottom-right corner of the image selection. The bottom coordinate is greater than the top coordinate.

As a result, the width of the image selection is equal to the right coordinate minus the left coordinate, and the height of the image selection is equal to the bottom coordinate minus the top coordinate.

The value of the attribute is a string in the 'left,top,right,bottom' pattern where: ',' is the U+002C COMMA character; and 'left', 'top', 'right', and 'bottom' are substrings representing respectively the left, top, right, and bottom coordinates. Each substring contains between one and four decimal characters (inclusive). Each decimal character of each substring is in the range from '0' to '9' (U+0030 DIGIT ZERO to U+0039 DIGIT NINE, inclusive). The substrings do not contain any leading '0' (U+0033 DIGIT ZERO) characters.

As a result, the value of the attribute does not contain any of the following characters: white space characters defined in Section 6, '+' (U+002B PLUS SIGN), '-' (U+002D HYPHEN-MINUS), and '.' (U+002E FULL STOP).

Examples of accepted values:
bounds='50,30,150,130'
bounds='250,30,450,130'

Examples of refused values:
bounds=''

```
bounds='50,30,1500,130'
bounds='150,30,50,130'
bounds='50,30'
bounds='50 30 150 130'
bounds='050,30,150,130'
bounds='50, 30,150,130'
bounds='+50,30,150,130'
bounds='50,-30,150,130'
bounds='50.5,30,150,130'
```

During the rendering of the Frogans slide, when preparing the image resource:

* If the left coordinate is greater than the width of the pre-authored image minus one, then the image selection is empty. In this case, the prepared image resource is fully transparent.

* If the top coordinate is greater than the height of the pre-authored image minus one, then the image selection is empty. In this case, the prepared image resource is fully transparent.

* If the right coordinate is greater than the width of the pre-authored image, then the right coordinate is considered to be equal to the width of the pre-authored image.

* If the bottom coordinate is greater than the height of the pre-authored image, then the bottom coordinate is considered to be equal to the height of the pre-authored image.

aspect - Optional

The aspect of the image selection in the image resource.

The attribute is based on a string set which represents the possible values of the attribute.

The value of the attribute is case-sensitive.

The value can be:

* 'base' indicates that the image selection is resized, maintaining its original proportions, such that the whole image selection occupies either the entire width or the entire height of the image resource. Any remaining parts of the image resource that are not occupied by the image selection are fully transparent. The placement of the image selection in the image resource depends on the value of the "adjust" attribute of the <resimage> element.

* 'spread' indicates that the image selection is resized, without maintaining its original proportions, such that the image selection occupies the entire width and height of the image resource.

* 'zoom' indicates that the image selection is resized, maintaining its original proportions, such that the image selection occupies the entire width and height of the image resource. The image selection is cropped in order to fit in the image resource if its proportions are different from the proportions of the image resource. The placement of the image selection in the image resource depends on the value of the "adjust" attribute of the <resimage> element.

* 'echo' indicates that the image selection is resized, maintaining its original proportions, such that the whole image selection occupies either the entire width or the entire height of the image resource. Any remaining parts of the image resource that are not occupied by the image selection are covered by copies of the image selection that are placed next to one another, either vertically or horizontally. Copies of the image selection are cropped in order to fit in the image resource if their placement makes them extend beyond the edge of the image resource. The placement of the image selection in the image resource depends on the value of the "adjust" attribute of the <resimage> element.

* 'tile' indicates that the image selection is not resized. Any remaining parts of the image resource that are not occupied by the image selection are covered by copies of the image selection that are placed next to one another, both vertically and horizontally. The image selection and copies of the image selection are cropped in order to fit in the image resource if their placement makes them extend beyond the edge of the image resource. The placement of the image selection in the image resource depends on the value of the "origin" attribute of the <resimage> element.

The default value is 'base'.

For more information on the process to resize image selections, see Appendix B.2.

adjust - Applicable only if the value of the "aspect" attribute equals 'base', 'zoom', or 'echo'. Optional if applicable.

The adjustment of the image selection in the image resource.

The value of the attribute represents an integer between -100 and 100 (inclusive).

The adjustment is applied, either horizontally or vertically, as follows:

* An adjustment of -100 corresponds to the LT placement, where the image selection is placed so that its left or top edge matches the left or top edge of the image resource.

* An adjustment of 0 corresponds to the central placement, where the image selection is centered vertically or horizontally in the image resource.

* An adjustment of 100 corresponds to the RB placement, where the image selection is placed so that its right or bottom edge matches the right or bottom edge of the image resource.

* An adjustment between -100 and 0 (exclusive) corresponds to a proportional placement, where the image selection is placed according to the absolute value of the adjustment, starting from the central placement and moving towards the LT placement. For instance, an adjustment of -25 corresponds to a placement where the image selection is placed one-quarter of the way from the central placement towards the LT placement.

* An adjustment between 0 and 100 (exclusive) corresponds to a proportional placement, where the image selection is placed according to the value of the adjustment, starting from the central placement and moving towards the RB placement. For instance, an adjustment of 75 corresponds to a placement where the image selection is placed three-quarters of the way from the central placement towards the RB placement.

The default value is 0.

The value of the attribute is a string in the 'pd' pattern where: 'p' is a substring representing the sign of the adjustment; and 'd' is a substring representing the absolute value of the adjustment. The 'p' substring is either an empty string, indicating that the value of the adjustment is greater than or equal to zero, or is the '-' (U+002D HYPHEN-MINUS) character. The 'd' substring contains between one and three decimal characters (inclusive). Each decimal character of the 'd' substring is in the range from '0' to '9' (U+0033 DIGIT ZERO to U+0039 DIGIT NINE, inclusive). The 'd' substring does not contain any leading '0'

(U+0033 DIGIT ZERO) characters.

As a result, the value of the attribute does not contain any of the following characters: white space characters defined in Section 6, '+' (U+002B PLUS SIGN), and '.' (U+002E FULL STOP).

Examples of accepted values:
adjust='0'
adjust='-100'
adjust='100'
adjust='-25'
adjust='75'

Examples of refused values:
adjust=''
adjust='-0'
adjust='-120'
adjust='150'
adjust='-025'
adjust='075'
adjust='- 25'
adjust='+100'
adjust='-25.5'

origin - Applicable only if the value of the "aspect" attribute equals 'tile'. Optional if applicable.

The origin in the image selection used in the image resource.

The value of the attribute represents the x and y coordinates, in pixels, of the origin in the image selection. The x and y coordinates correspond to the horizontal and vertical coordinates, respectively, of the pixel in the image selection that is placed in the top-left corner of the image resource.

The x and y coordinates are integers between 0 and 1,023 (inclusive).

The x and y coordinates of the top-left corner of the image selection are both zero.

The default value is '0,0'.

The value of the attribute is a string in the 'x,y' pattern where: ',' is the U+002C COMMA character; and 'x' and 'y' are substrings representing respectively the x and y coordinates. Each substring contains between one and four decimal characters (inclusive). Each decimal character of each substring is in the range from '0'

to '9' (U+0033 DIGIT ZERO to U+0039 DIGIT NINE, inclusive). The substrings do not contain any leading '0' (U+0033 DIGIT ZERO) characters.

As a result, the value of the attribute does not contain any of the following characters: white space characters defined in Section 6, '+' (U+002B PLUS SIGN), '-' (U+002D HYPHEN-MINUS), and '.' (U+002E FULL STOP).

Examples of accepted values:
origin='0,0'
origin='50,30'
origin='250,30'

Examples of refused values:
origin=''
origin='50,3000'
origin='50'
origin='50 30'
origin='050,30'
origin='50, 30'
origin='+50,30'
origin='50,-30'
origin='50.5,30'

During the rendering of the Frogans slide, when preparing the image resource:

* If the x coordinate is greater than the width of the pre-authored image minus one, then the x coordinate is considered to be equal to the width of the pre-authored image minus one.

* If the y coordinate is greater than the height of the pre-authored image minus one, then the y coordinate is considered to be equal to the height of the pre-authored image minus one.

7.3. Preparing Pixels Resources - <respixels> Element

The <respixels> element prepares a pixels resource that is based on a miniature bitmap.

The <respixels> element is a child of the <frogans-fsdl> element (see Section 6.1). The <respixels> element does not contain any children. It contains text characters.

As stated in the Frogans slide design concepts (Section 4):

- The pixels resource is intended to be used in the Frogans slide as the foundation of a layer (see the <layer> element in Section 7.12) or as part of a merge resource (see the <resmerge> and <merge> elements in Section 7.11).

- The pixels resource can be used once or several times in the Frogans slide.

The pixels resource is identified using a component identifier (Section 6) which is determined by the value of the "resid" attribute of the <respixels> element.

The size, in pixels, of the pixels resource depends on the value of the "size" attribute of the <respixels> element.

The miniature bitmap is made up of individual pixels, organized in columns and rows.

The miniature bitmap is resized, without maintaining its original proportions, such that the miniature bitmap occupies the entire width and height of the pixels resource.

The pixels resource can be used, for instance, as a background with color gradients, or as a mask with transparency gradients.

For more information on the process to resize miniature bitmaps, see Appendix B.3.

The number of columns of the miniature bitmap depends on the value of the "columns" attribute of the <respixels> element. The number of rows of the miniature bitmap depends on the value of the "rows" attribute of the <respixels> element.

Each individual pixel of the miniature bitmap is defined using four components: the R, G and B components (color) and the A component (transparency).

The R, G, B, and A components of the individual pixels of the miniature bitmap are all defined in the same way, depending on the value of the "pix" attribute of the <respixels> element.

The definition of all the individual pixels of the miniature bitmap can be either:

* an rgba definition, where, for an individual pixel, the values of the R, G, B, and A components are defined in the text characters of the <respixels> element.

* an rgb definition, where, for an individual pixel, the values of the R, G, and B components are defined in the text characters of the <respixels> element, and the value of the A component depends on the value of the "alpha" attribute of the <respixels> element.

* an a definition, where, for an individual pixel, the values of the R, G, and B components depend on the value of the "color" attribute of the <respixels> element, and the value of the A component is defined in the text characters of the <respixels> element.

* a y definition, where, for an individual pixel, the values of the R, G, and B components are defined in the text characters of the <respixels> element, and the value of the A component depends on the value of the "alpha" attribute of the <respixels> element. Using this definition, the R, G, and B components share the same value, corresponding to the luminance of the individual pixel, hence the color of the individual pixel is on a gray scale.

* a ya definition, where, for an individual pixel, the R, G, B and A components are defined in the text characters of the <respixels> element. Using this definition, the R, G, and B components share the same value, corresponding to the luminance of the individual pixel, hence the color of the individual pixel is on a gray scale.

The text characters of the <respixels> element contain a list of items. Each item corresponds to an individual pixel.

The individual pixels are ordered row by row, from the top-left to the bottom-right pixel of the miniature bitmap. The number of items in the list is the product of the number of rows and the number of columns.

All items in the list are strings using one of the following patterns:

* '#rrggbbaa': this pattern corresponds to the rgba definition.

* '#rrggbb': this pattern corresponds to the rgb definition.

* '#aa': this pattern corresponds to the a definition.

* '#yy': this pattern corresponds to the y definition.

* '#yyaa': this pattern corresponds to the ya definition.

In the patterns above: '#' is the U+0013 NUMBER SIGN character; 'rr', 'gg', and 'bb' are substrings representing respectively the R, G, and B components; 'aa' is a substring representing the A component; and 'yy' is a substring representing the value shared by the R, G, and B components (luminance). Each substring contains two hexadecimal characters. Each hexadecimal character of each substring is in one of the following ranges: from '0' to '9' (U+0030 DIGIT ZERO to U+0039 DIGIT NINE, inclusive), from 'A' to 'F' (U+0041 LATIN CAPITAL LETTER A to U+0046 LATIN CAPITAL LETTER F, inclusive), from 'a' to 'f' (U+0061 LATIN SMALL LETTER A to U+0066 LATIN SMALL LETTER F, inclusive).

As a result, the pattern does not contain any white space characters defined in Section 6.

Examples of accepted items:
#004698bf
#004698
#bf
#3a
#3abf
#3Abf
#3ABF

Examples of refused items:
004698bf
#
#004698b
#00469
#b
#3ab
004698bf
#004698 bf
#0004698bf

The separator character used to separate items in the list is the ';' (U+003B SEMICOLON) character.

White space characters, as defined in Section 6, can optionally be included before and after the ';' (U+003B SEMICOLON) separator characters in the list. These white space characters are ignored.

The list cannot contain empty items. As a result, there is no separator character either before the first item or after the last item in the list. Furthermore, there cannot be two consecutive separator characters in the list.

Examples of accepted item lists:
#004698bf;#ffbf00ff;#00469840
#004698;#ffbf00;#001c46
#bf;#ff
#3abf;#00ff;#3abf;#ff00

Examples of refused item lists:
#004698bf;#ffbf00;#00469840
#004698;#bf;#3abf
#004698bf#ffbf00ff
#004698bf #ffbf00ff
;#004698bf;#ffbf00ff
#004698bf;#ffbf00ff;
#004698bf;;#ffbf00ff

During the rendering of the Frogans slide, when preparing the pixels resource, individual pixels of the miniature bitmap can be invisible pixels, i.e. the level of the A component (transparency) is equal to zero. In this case, as stated in the Frogans slide design concepts, the levels of the R, G and B components (color) of the individual pixel are considered to be equal to zero.

The <respixels> element has the following attributes:

resid - Mandatory

The identifier of the pixels resource.

The value of the attribute is a component identifier. The value is compliant with the rules presented in Section 6.

As a result, the value cannot have been utilized previously in the FSDL document as a component identifier.

Examples of accepted values:
resid='foo1'
resid='Foo1'

```
        resid='foo2'
        resid='foo_1'

        Examples of refused values:
        resid=''
        resid='foo-1'
        resid='foo/1'
        resid='foo.1'
        resid='foo 1'
        resid='foo1,foo2'
```

size - Mandatory

The size of the pixels resource.

The value of the attribute represents the width and height, in pixels, of the pixels resource.

As stated in the Frogans slide design concepts (Section 4), the width of the pixels resource is an integer between 1 and 640 (inclusive) and its height is an integer between 1 and 480 (inclusive).

The value of the attribute is a string in the 'width,height' pattern where: ',' is the U+002C COMMA character; and 'width' and 'height' are substrings representing respectively the width and height of the pixels resource. Each substring contains between one and three decimal characters (inclusive). Each decimal character of the substring is in the range from '0' to '9' (U+0033 DIGIT ZERO to U+0039 DIGIT NINE, inclusive). The substring does not contain any leading '0' (U+0033 DIGIT ZERO) characters.

As a result, the value of the attribute does not contain any of the following characters: white space characters defined in Section 6, '+' (U+002B PLUS SIGN), '-' (U+002D HYPHEN-MINUS), and '.' (U+002E FULL STOP).

```
   Examples of accepted values:
   size='40,200'
   size='400,200'

   Examples of refused values:
   size=''
   size='400,0'
   size='400,550'
   size='400'
```

```
size='400 200'
size='040,200'
size='400, 200'
size='+400,200'
size='400,-200'
size='400.5,200'
```

columns - Mandatory

The number of columns of the miniature bitmap on which the pixels resource is based.

The value of the attribute represents an integer between 1 and 16 (inclusive).

The value of the attribute is a string containing between one and two decimal characters (inclusive). Each decimal character of the string is in the range from '0' to '9' (U+0030 DIGIT ZERO to U+0039 DIGIT NINE, inclusive). The string does not contain any leading '0' (U+0033 DIGIT ZERO) characters.

As a result, the value of the attribute does not contain any of the following characters: white space characters defined in Section 6, '+' (U+002B PLUS SIGN), '-' (U+002D HYPHEN-MINUS), and '.' (U+002E FULL STOP).

Examples of accepted values:

```
columns='1'
columns='16'
columns='9'
```

Examples of refused values:

```
columns=''
columns='-9'
columns='20'
columns='09'
columns='+9'
columns='9.5'
```

rows - Mandatory

The number of rows of the miniature bitmap on which the pixels resource is based.

The value of the attribute represents an integer between 1 and 16 (inclusive).

The value of the attribute is a string containing between one and two decimal characters (inclusive). Each decimal character of the string is in the range from '0' to '9' (U+0030 DIGIT ZERO to U+0039 DIGIT NINE, inclusive). The string does not contain any leading '0' (U+0033 DIGIT ZERO) characters.

As a result, the value of the attribute does not contain any of the following characters: white space characters defined in Section 6, '+' (U+002B PLUS SIGN), '-' (U+002D HYPHEN-MINUS), and '.' (U+002E FULL STOP).

Examples of accepted values:
rows='1'
rows='16'
rows='9'

Examples of refused values:
rows=''
rows='-9'
rows='20'
rows='09'
rows='+9'
rows='9.5'

pix - Mandatory

The way of defining all the individual pixels of the miniature bitmap on which the pixels resource is based.

The attribute is based on a string set which represents the possible values of the attribute.

The value of the attribute is case-sensitive.

The value can be:

* 'rgba' indicates that all the individual pixels of the miniature bitmap use an rgba definition.

* 'rgb' indicates that all the individual pixels of the miniature bitmap use an rgb definition.

* 'a' indicates that all the individual pixels of the miniature bitmap use an a definition.

* 'y' indicates that all the individual pixels of the miniature bitmap use a y definition.

* 'ya' indicates that all the individual pixels of the miniature bitmap use a ya definition.

color - Applicable only if the value of the "pix" attribute equals 'a'. Optional if applicable.

The color of the individual pixels of the miniature bitmap on which the pixels resource is based.

The value of the attribute represents the levels of the R, G, and B components of the color. The level of each component is represented using two hexadecimal digits.

As stated in the Frogans slide design concepts (Section 4), the level of each one of the R, G, and B components is in the range from 00 (0) to ff (255).

The default value is '#0000ff'.

The value of the attribute is a string in the '#rrggbb' pattern where: '#' is the U+0013 NUMBER SIGN character; and 'rr', 'gg', and 'bb' are substrings representing respectively the R, G, and B components. Each substring contains two hexadecimal characters. Each hexadecimal character of each substring is in one of the following ranges: from '0' to '9' (U+0030 DIGIT ZERO to U+0039 DIGIT NINE, inclusive), from 'A' to 'F' (U+0041 LATIN CAPITAL LETTER A to U+0046 LATIN CAPITAL LETTER F, inclusive), from 'a' to 'f' (U+0061 LATIN SMALL LETTER A to U+0066 LATIN SMALL LETTER F, inclusive).

As a result, the value of the attribute does not contain any white space characters defined in Section 6.

Examples of accepted values:
color='#0000ff'
color='#004698'
color='#001c46'
color='#ffbf00'
color='#Ffbf00'
color='#FFBF00'

Examples of refused values:
color=''
color='ffbf00'
color='#'
color='#ffbf'

```
color='# ffbf00'
color='#ffbf 00'
color='#0ffbf00'
color='#00f'
```

alpha - Applicable only if the value of the "pix" attribute equals either 'rgb' or 'y'. Optional if applicable.

The transparency of the individual pixels of the miniature bitmap on which the pixels resource is based.

The value of the attribute represents the level of the A component of the individual pixels. The level of the A component is represented using two hexadecimal digits.

As stated in the Frogans slide design concepts, the level of the A component is in the range from 00 (0) to ff (255), where the level 00 corresponds to a fully transparent pixel and ff corresponds to a fully opaque pixel.

The default value is '#ff'.

The value of the attribute is a string in the '#aa' pattern where: '#' is the U+0013 NUMBER SIGN character; and 'aa' is a substring representing the A component. Each substring contains two hexadecimal characters. Each hexadecimal character of each substring is in one of the following ranges: from '0' to '9' (U+0030 DIGIT ZERO to U+0039 DIGIT NINE, inclusive), from 'A' to 'F' (U+0041 LATIN CAPITAL LETTER A to U+0046 LATIN CAPITAL LETTER F, inclusive), from 'a' to 'f' (U+0061 LATIN SMALL LETTER A to U+0066 LATIN SMALL LETTER F, inclusive).

As a result, the value of the attribute does not contain any white space characters defined in Section 6.

Examples of accepted values:

```
alpha='#ff'
alpha='#00'
alpha='#bf'
alpha='#Bf'
alpha='#BF'
```

Examples of refused values:

```
alpha=''
alpha='bf'
alpha='#'
```

```
alpha='#b'
alpha='# bf'
alpha='#b f'
alpha='#0bf'
```

7.4. Preparing Drawing Resources - <resdraw> Element

The <resdraw> element prepares a drawing resource that is based on a geometric form.

The <resdraw> element is a child of the <frogans-fsdl> element (see Section 6.1). The <resdraw> element does not contain any children. It does not contain text characters.

As stated in the Frogans slide design concepts (Section 4):

- The drawing resource is intended to be used in the Frogans slide as the foundation of a layer (see the <layer> element in Section 7.12) or as part of a merge resource (see the <resmerge> and <merge> elements in Section 7.11).

- The drawing resource can be used once or several times in the Frogans slide.

The drawing resource is identified using a component identifier (Section 6) which is determined by the value of the "resid" attribute of the <resdraw> element.

The size, in pixels, of the drawing resource depends on the value of the "size" attribute of the <resdraw> element.

The geometric form is either a rectangle, a rounded rectangle, or an ellipse, depending on the value of the "figure" attribute of the <resdraw> element.

The geometric form is drawn either as a line or as a filled area, depending on the value of the "stroke" attribute of the <resdraw> element. If the geometric form is drawn as a line, then the thickness of the line depends on the value of the "thick" attribute of the <resdraw> element.

The geometric form covers the entire width and height of the drawing resource.

If the geometric form is drawn as a line, then the thickness of the line is fully included in the drawing resource, i.e. it does not extend beyond the edge of the drawing resource.

A rounded rectangle is a rectangle with four rounded corners, where each corner is one quarter of an ellipse. For a rounded rectangle:

* The width and height of the ellipse used to draw the corners depend on the value of the "round" attribute of the <resdraw>

element.

* The top-left, top-right, bottom-left, and bottom-right corners correspond respectively to the top-left, top-right, bottom-left, and bottom-right quarters of the ellipse.

* If the width of the ellipse is equal to the width of the drawing resource, then the top and bottom of the rounded rectangle are both a semi-ellipse.

* If the height of the ellipse is equal to the height of the drawing resource, then the left and right sides of the rounded rectangle are both a semi-ellipse.

* If the width and height of the ellipse are equal respectively to the width and height of the drawing resource, then the rounded rectangle becomes an ellipse.

The color of the drawing resource depends on the value of the "color" attribute of the <resdraw> element.

For more information on the process to draw geometric forms, see Appendix B.4.

The <resdraw> element has the following attributes:

resid - Mandatory

The identifier of the drawing resource.

The value of the attribute is a component identifier. The value is compliant with the rules presented in Section 6.

As a result, the value cannot have been utilized previously in the FSDL document as a component identifier.

Examples of accepted values:
resid='foo1'
resid='Foo1'
resid='foo2'
resid='foo_1'

Examples of refused values:
resid=''
resid='foo-1'
resid='foo/1'

```
        resid='foo.1'
        resid='foo 1'
        resid='foo1,foo2'
```

size - Mandatory

The size of the drawing resource.

The value of the attribute represents the width and height, in pixels, of the drawing resource.

As stated in the Frogans slide design concepts (Section 4), the width of the drawing resource is an integer between 1 and 640 (inclusive) and its height is an integer between 1 and 480 (inclusive).

The value of the attribute is a string in the 'width,height' pattern where: ',' is the U+002C COMMA character; and 'width' and 'height' are substrings representing respectively the width and height of the drawing resource. Each substring contains between one and three decimal characters (inclusive). Each decimal character of the substring is in the range from '0' to '9' (U+0033 DIGIT ZERO to U+0039 DIGIT NINE, inclusive). The substring does not contain any leading '0' (U+0033 DIGIT ZERO) characters.

As a result, the value of the attribute does not contain any of the following characters: white space characters defined in Section 6, '+' (U+002B PLUS SIGN), '-' (U+002D HYPHEN-MINUS), and '.' (U+002E FULL STOP).

```
        Examples of accepted values:
        size='40,200'
        size='400,200'

        Examples of refused values:
        size=''
        size='400,0'
        size='400,550'
        size='400'
        size='400 200'
        size='040,200'
        size='400, 200'
        size='+400,200'
        size='400,-200'
        size='400.5,200'
```

figure - Mandatory

The geometric form on which the drawing resource is based.

The attribute is based on a string set which represents the possible values of the attribute.

The value of the attribute is case-sensitive.

The value can be:

* 'rect' indicates that the geometric form is a rectangle.

* 'roundrect' indicates that the geometric form is a rounded rectangle.

* 'ellipse' indicates that the geometric form is an ellipse. If the width and height of the drawing resource are equal, then the geometric form is a circle.

stroke - Mandatory

The way of drawing the geometric form on which the drawing resource is based.

The attribute is based on a string set which represents the possible values of the attribute.

The value of the attribute is case-sensitive.

The value can be:

* 'on' indicates that the geometric form is drawn as a line. The thickness of the line depends on the value of the "thick" attribute of the <resdraw> element.

* 'off' indicates that the geometric form is drawn as a filled area.

thick - Applicable only if the value of the "stroke" attribute equals 'on'. Optional if applicable.

The thickness of the line in the drawing resource.

The value of the attribute represents an integer between 1 and 64 (inclusive).

The thickness is in pixels.

The default value is '8'.

The value of the attribute is a string containing between one and two decimal characters (inclusive). Each decimal character of the string is in the range from '0' to '9' (U+0030 DIGIT ZERO to U+0039 DIGIT NINE, inclusive). The string does not contain any leading '0' (U+0033 DIGIT ZERO) characters.

As a result, the value of the attribute does not contain any of the following characters: white space characters defined in Section 6, '+' (U+002B PLUS SIGN), '-' (U+002D HYPHEN-MINUS), and '.' (U+002E FULL STOP).

Examples of accepted values:
thick='1'
thick='64'
thick='9'

Examples of refused values:
thick=''
thick='-9'
thick='80'
thick='+9'
thick='9.5'

round - Applicable only if the value of the "figure" attribute equals 'roundrect'. Optional if applicable.

The dimensions of the ellipse used to draw the corners of the rounded rectangle on which the drawing resource is based.

The value of the attribute represents the width and height, in pixels, of the ellipse.

The width of the ellipse is an integer between 1 and 640 (inclusive) and its height is an integer between 1 and 480 (inclusive).

The default value is '16,16'.

The value of the attribute is a string in the 'width,height' pattern where: ',' is the U+002C COMMA character; and 'width' and 'height' are substrings representing respectively the width and height of the ellipse. Each substring contains between one and three decimal characters (inclusive). Each decimal character of each substring is in the range from '0' to '9' (U+0033 DIGIT ZERO to U+0039 DIGIT NINE, inclusive). The substrings do not contain any leading '0' (U+0033 DIGIT ZERO) characters.

As a result, the value of the attribute does not contain any of the following characters: white space characters defined in Section 6, '+' (U+002B PLUS SIGN), '-' (U+002D HYPHEN-MINUS), and '.' (U+002E FULL STOP).

Examples of accepted values:
origin='16,16'
origin='40,200'
origin='400,200'

Examples of refused values:
origin=''
origin='400,0'
origin='400,550'
origin='400'
origin='400 200'
origin='040,200'
origin='400, 200'
origin='+400,200'
origin='400,-200'
origin='400.5,200'

During the rendering of the Frogans slide, when preparing the drawing resource:

* If the width of the ellipse is greater than the width of the drawing resource, then the width of the ellipse is considered to be equal to the width of the drawing resource.

* If the height of the ellipse is greater than the height of the drawing resource, then the height of the ellipse is considered to be equal to the height of the drawing resource.

color - Optional

The color of the drawing resource.

The value of the attribute represents the levels of the R, G, and B components of the color. The level of each component is represented using two hexadecimal digits.

As stated in the Frogans slide design concepts (Section 4), the level of each one of the R, G, and B components is in the range from 00 (0) to ff (255).

The default value is '#0000ff'.

The value of the attribute is a string in the '#rrggbb' pattern where: '#' is the U+0013 NUMBER SIGN character; and 'rr', 'gg', and 'bb' are substrings representing respectively the R, G, and B components. Each substring contains two hexadecimal characters. Each hexadecimal character of each substring is in one of the following ranges: from '0' to '9' (U+0030 DIGIT ZERO to U+0039 DIGIT NINE, inclusive), from 'A' to 'F' (U+0041 LATIN CAPITAL LETTER A to U+0046 LATIN CAPITAL LETTER F, inclusive), from 'a' to 'f' (U+0061 LATIN SMALL LETTER A to U+0066 LATIN SMALL LETTER F, inclusive).

As a result, the value of the attribute does not contain any white space characters defined in Section 6.

```
Examples of accepted values:
color='#0000ff'
color='#004698'
color='#001c46'
color='#ffbf00'
color='#Ffbf00'
color='#FFBF00'

Examples of refused values:
color=''
color='ffbf00'
color='#'
color='#ffbf'
color='# ffbf00'
color='#ffbf 00'
color='#0ffbf00'
color='#00f'
```

7.5. Preparing Path Resources - <respath> Element

The <respath> element prepares a path resource that is based on a plane containing Bezier curves.

The <respath> element is a child of the <frogans-fsdl> element (see Section 6.1). The <respath> element does not contain any children. It contains text characters.

As stated in the Frogans slide design concepts (Section 4):

- The path resource is intended to be used in the Frogans slide as the foundation of a layer (see the <layer> element in Section 7.12) or as part of a merge resource (see the <resmerge> and <merge> elements in Section 7.11).

- The path resource can be used once or several times in the Frogans slide.

The path resource is identified using a component identifier (Section 6) which is determined by the value of the "resid" attribute of the <respath> element.

The size, in pixels, of the path resource depends on the value of the "size" attribute of the <respath> element.

Each Bezier curve contained in the plane is either:

* a linear Bezier curve, which has a starting point and an ending point.

* a conic Bezier curve, which has a starting point, an ending point, and one control point. A conic Bezier curve is also referred to as a quadratic Bezier curve.

* a cubic Bezier curve, which has a starting point, an ending point, and two control points.

Each Bezier curve can be used in the plane either as:

* a simple Bezier curve, meaning that it is independent from all the other Bezier curves contained in the plane, or

* a part of a composite Bezier curve. A composite Bezier curve is made up of two of more linear, conic, or cubic Bezier curves, joined end to end in any combination, such that the ending point of one curve coincides with the starting point of the next curve.

The plane can contain any combination of simple and composite Bezier curves.

In order to position the starting, ending, and control points of the simple and composite Bezier curves in the plane, the plane uses an x,y coordinate system where: the x and the y coordinates of a point in the plane represent the distances of the point from the x and y coordinate axes; the x coordinate is an integer which increases towards the right from 0 to 2,048; and the y coordinate is an integer which increases towards the bottom from 0 to 2,048.

The plane can be cropped in order to select which part of the plane is apparent in the path resource, depending on the values of the "crop" and "corners" attributes of the <respath> element.

For instance, the plane can be cropped automatically in cases where the x,y coordinates of the starting, ending, and control points of the Bezier curves contained in the plane do not use the full range from 0 to 2,048. Furthermore, the plane can be cropped using custom coordinates in order to exclude certain parts or control points of Bezier curves, or in order to draw, on a fixed scale, graphs representing data that is generated on the fly (i.e. dynamically) on the server hosting the Frogans site.

All the simple and composite Bezier curves are drawn either as lines or as filled areas, depending on the value of the "stroke" attribute of the <respath> element. If the simple and composite Bezier curves are drawn as lines, then the thickness of the lines depends on the value of the "thick" attribute of the <respath> element.

If the simple and composite Bezier curves are drawn as lines, then they can all be closed depending on the value of the "close" attribute of the <respath> element.

If a simple Bezier curve is closed, then a straight line is drawn from its ending point to its starting point.

If a composite Bezier curve is closed, then a straight line is drawn from the ending point of its last Bezier curve to the starting point of its first Bezier curve. A composite Bezier curve cannot be closed correctly by appending a linear Bezier curve whose ending point is the starting point of the first Bezier curve of the composite Bezier curve.

If the simple and composite Bezier curves are drawn as filled areas, then the rule used to determine these filled areas depends on the value of the "fill" attribute of the <respath> element.

The way in which the apparent part of the plane is mapped to the path resource depends on the values of the "spread" and "adjust" attributes of the <respath> element. The size, in x,y coordinates, of the apparent part of the plane is not necessarily the same as the size, in pixels, of the path resource.

If the simple and composite Bezier curves are drawn as lines, and if their starting, ending, and control points are all included in the apparent part of the plane, then the thickness of the lines is fully included in the path resource, i.e. it does not extend beyond the edge of the path resource.

The color of the path resource depends on the value of the "color" attribute of the <respath> element.

The text characters of the <respath> element contain a list of items. Each item can be either:

* a jump item, which corresponds to the starting point of a simple or composite Bezier curve

* a linear item, which corresponds to the ending point of a linear Bezier curve

* a conic item, which corresponds to the ending point and the control point of a conic Bezier curve, or

* a cubic item, which corresponds to the ending point and both control points of a cubic Bezier curve

Each simple and composite Bezier curve contained in the plane is represented as a sequence of items in the list:

* A simple Bezier curve is represented as a jump item followed by one linear, conic, or cubic item.

* A composite Bezier curve is represented as a jump item followed by any combination of two or more linear, conic, or cubic items.

The simple and composite Bezier curves contained in the plane are included one after another in the list.

As a result, the list starts with a jump item, cannot end with a jump item, and cannot contain two consecutive jump items.

Each item in the list is a string using one of the following patterns:

* 'Ju:xstart,ystart': this pattern corresponds to a jump item.

* 'Li:xend,yend': this pattern corresponds to a linear item.

* 'Co:xend,yend,xcp1,ycp1': this pattern corresponds to a conic item.

* 'Cu:xend,yend,xcp1,ycp1,xcp2,ycp2': this pattern corresponds to a cubic item.

In the patterns above: 'Ju', 'Li', 'Co', and 'Cu' are case-sensitive string constants representing the kind of item; ':' is the U+003A COLON character; ',' is the U+002C COMMA character; 'xstart' and 'ystart' are substrings representing respectively the x and y coordinates of a starting point; 'xend' and 'yend' are substrings representing respectively the x and y coordinates of an ending point; 'xcp1' and 'ycp1' are substrings representing respectively the x and y coordinates of either the control point of a conic Bezier curve, or the first control point of a cubic Bezier curve; and 'xcp2' and 'ycp2' are substrings representing respectively the x and y coordinates of the second control point of a cubic Bezier curve. Each substring contains between one and four decimal characters (inclusive). Each decimal character of each substring is in the range from '0' to '9' (U+0030 DIGIT ZERO to U+0039 DIGIT NINE, inclusive). The substrings do not contain any leading '0' (U+0033 DIGIT ZERO) characters.

As a result, the item does not contain any of the following characters: white space characters defined in Section 6, '+' (U+002B PLUS SIGN), '-' (U+002D HYPHEN-MINUS), and '.' (U+002E FULL STOP).

Examples of accepted items:

```
Ju:200,1000
Li:200,1000
Co:200,1000,450,1300
Cu:200,1000,450,1300,1850,300
```

Examples of refused items:

```
200,1000
J200,1000
Ju200,1000
Ju 200,1000
Ju=200,1000
ju:200,1000
JU:200,1000
Ju:200,2400
```

```
Ju:200
Ju:200 1000
Ju:0200,1000
Ju:200, 1000
Ju:+200,1000
Ju:200,-1000
Ju:200.5,1000
Li:200
Co:200,1000
Co:200,1000,450
Cu:200,1000,450,1300
Cu:200,1000,450,1300,1850
```

The separator character used to separate items in the list is the ';' (U+003B SEMICOLON) character.

White space characters, as defined in Section 5.5, can optionally be included before and after the ';' (U+003B SEMICOLON) separator characters in the list. These white space characters are ignored.

The list cannot contain empty items. As a result, there is no separator character either before the first item or after the last item in the list. Furthermore, there cannot be two consecutive separator characters in the list.

The number of items in the list is between 2 and 512 (inclusive).

Examples of accepted item lists:

```
Ju:10,50;Li:200,1000
Ju:10,50;Co:200,1000,450,1300
Ju:10,50;Cu:200,1000,450,1300,1850,300
Ju:10,50;Li:500,100;Li:200,1000
Ju:10,50;Li:500,100;Co:200,1000,450,1300
Ju:10,50;Li:500,100;Cu:200,1000,450,1300,1850,300
Ju:10,50;Li:75,1400;Ju:1400,1600;Li:500,100
Ju:10,50;Li:75,1400;Ju:1400,1600;Li:500,100;Co:200,1000,450,1300
```

Examples of refused item lists:

```
Li:75,1400;Ju:1400,1600;Li:500,100
Ju:10,50;Li:75,1400;Ju:1400,1600
Ju:10,50;Li:75,1400;Ju:200,1000;Ju:1400,1600;Li:500,100
Ju:10,50Li:200,1000
Ju:10,50 Li:200,1000
;Ju:10,50;Li:200,1000
Ju:10,50;Li:200,1000;
Ju:10,50;;Li:200,1000
```

For more information on the process to draw Bezier curves, see

Appendix B.5.

The <respath> element has the following attributes:

resid - Mandatory

The identifier of the path resource.

The value of the attribute is a component identifier. The value is compliant with the rules presented in Section 6.

As a result, the value cannot have been utilized previously in the FSDL document as a component identifier.

Examples of accepted values:
resid='foo1'
resid='Foo1'
resid='foo2'
resid='foo_1'

Examples of refused values:
resid=''
resid='foo-1'
resid='foo/1'
resid='foo.1'
resid='foo 1'
resid='foo1,foo2'

size - Mandatory

The size of the path resource.

The value of the attribute represents the width and height, in pixels, of the path resource.

As stated in the Frogans slide design concepts (Section 4), the width of the path resource is an integer between 1 and 640 (inclusive) and its height is an integer between 1 and 480 (inclusive).

The value of the attribute is a string in the 'width,height' pattern where: ',' is the U+002C COMMA character; and 'width' and 'height' are substrings representing respectively the width and height of the path resource. Each substring contains between one and three decimal characters (inclusive). Each decimal character of the substring is in the range from '0' to '9' (U+0033 DIGIT ZERO to U+0039 DIGIT NINE, inclusive). The substring does not contain any leading '0' (U+0033 DIGIT ZERO) characters.

As a result, the value of the attribute does not contain any of the following characters: white space characters defined in Section 6, '+' (U+002B PLUS SIGN), '-' (U+002D HYPHEN-MINUS), and '.' (U+002E FULL STOP).

Examples of accepted values:
size='40,200'
size='400,200'

Examples of refused values:
size=''
size='400,0'
size='400,550'
size='400'
size='400 200'
size='040,200'
size='400, 200'
size='+400,200'
size='400,-200'
size='400.5,200'

crop - Mandatory

The cropping applied to the plane on which the path resource is based.

The attribute is based on a string set which represents the possible values of the attribute.

The value of the attribute is case-sensitive.

The value can be:

* 'none' indicates that no cropping is applied to the plane. The x and y coordinates of the point at the top-left corner of the apparent part of the plane are both zero. The x and y coordinates of the point at the bottom-right corner of the apparent part of the plane are both 2,048.

* 'auto' indicates that the plane is cropped automatically. The x and y coordinates of the point at the top-left corner of the apparent part of the plane are xmin and ymin, where xmin is the smallest x coordinate and ymin is the smallest y coordinate of all the starting, ending, and control points of all the Bezier curves contained in the plane. The x and y coordinates of the point at the bottom-right corner of the apparent part of the plane are xmax and ymax, where xmax is the largest x coordinate and ymax is the largest y coordinate of all the starting,

ending, and control points of all the Bezier curves contained in the plane.

* 'custom' indicates that the plane is cropped using custom coordinates. The x,y coordinates of the points at the top-left and bottom-right corners of the apparent part of the plane depend on the value of the "corners" attribute of the <respath> element.

corners - Applicable only if the value of the "crop" attribute equals 'custom'. Mandatory if applicable.

The top-left and bottom-right corners of the apparent part of the plane on which the path resource is based.

The value of the attribute represents the x and y coordinates of the points at the top-left and bottom-right corners of the apparent part of the plane.

The x and y coordinates of the point at the top-left corner are integers between 0 and 2,047 (inclusive). The x and y coordinates of the point at the bottom-right corner are integers between 1 and 2,048 (inclusive).

The x and y coordinates of the point at the bottom-right corner are greater respectively than the x and y coordinates of the point at the top-left corner.

The value of the attribute is a string in the 'xtl,ytl,xbr,ybr' pattern where: ',' is the U+002C COMMA character; and 'xtl' and 'ytl' are substrings representing respectively the x and y coordinates of the point at the top-left corner and 'xbr' and 'ybr' are substrings representing respectively the x and y coordinates of the point at the bottom-right corner. Each substring contains between one and four decimal characters (inclusive). Each decimal character of each substring is in the range from '0' to '9' (U+0030 DIGIT ZERO to U+0039 DIGIT NINE, inclusive). The substrings do not contain any leading '0' (U+0033 DIGIT ZERO) characters.

As a result, the value of the attribute does not contain any of the following characters: white space characters defined in Section 6, '+' (U+002B PLUS SIGN), '-' (U+002D HYPHEN-MINUS), and '.' (U+002E FULL STOP).

Examples of accepted values:

```
corners='10,50,200,1000'
corners='200,50,450,1000'

Examples of refused values:
corners=''
corners='10,50,2400,1000'
corners='450,50,200,1000'
corners='10,50'
corners='10 50 200 1000'
corners='010,50,200,1000'
corners='10, 50,200,1000'
corners='+10,50,200,1000'
corners='10,-50,200,1000'
corners='10.5,50,200,1000'
```

stroke - Mandatory

The way of drawing all the simple and composite Bezier curves contained in the plane on which the path resource is based.

The attribute is based on a string set which represents the possible values of the attribute.

The value of the attribute is case-sensitive.

The value can be:

* 'on' indicates that all the simple and composite Bezier curves are drawn as lines. The thickness of the lines depends on the value of the "thick" attribute of the <respath> element.

* 'off' indicates that all the simple and composite Bezier curves are drawn as filled areas.

thick - Applicable only if the value of the "stroke" attribute equals 'on'. Optional if applicable.

The thickness of the lines in the path resource.

The value of the attribute represents an integer between 1 and 64 (inclusive).

The thickness is in pixels.

The default value is '8'.

The value of the attribute is a string containing between one and two decimal characters (inclusive). Each decimal character of the string is in the range from '0' to '9' (U+0030 DIGIT ZERO to U+0039 DIGIT NINE, inclusive). The string does not contain any leading '0' (U+0033 DIGIT ZERO) characters.

As a result, the value of the attribute does not contain any of the following characters: white space characters defined in Section 6, '+' (U+002B PLUS SIGN), '-' (U+002D HYPHEN-MINUS), and '.' (U+002E FULL STOP).

Examples of accepted values:
thick='1'
thick='64'
thick='9'

Examples of refused values:
thick=''
thick='-9'
thick='80'
thick='+9'
thick='9.5'

close - Applicable only if the value of the "stroke" attribute equals 'on'. Optional if applicable.

Whether to close all the simple and composite Bezier curves contained in the plane on which the path resource is based.

The attribute is based on a string set which represents the possible values of the attribute.

The value of the attribute is case-sensitive.

The value can be:

* 'on' indicates that all the simple and composite Bezier curves are closed.

* 'off' indicates that none of the simple and composite Bezier curves are closed.

The default value is 'off'.

fill - Applicable only if the value of the "stroke" attribute equals 'off'. Optional if applicable.

The rule used to determine the filled areas in the path resource.

The attribute is based on a string set which represents the possible values of the attribute.

The value of the attribute is case-sensitive.

The value can be:

* 'non-zero' indicates that the non-zero winding rule is used.

* 'even-odd' indicates that the even-odd fill rule is used.

The default value is 'non-zero'.

spread - Mandatory

Whether the apparent part of the plane is spread when mapped to the path resource.

The attribute is based on a string set which represents the possible values of the attribute.

The value of the attribute is case-sensitive.

The value can be:

* 'on' indicates that the apparent part of the plane is mapped to the path resource, without maintaining its original proportions, such that the apparent part of the plane covers the entire width and height of the path resource.

* 'off' indicates that the apparent part of the plane is mapped to the path resource, maintaining its original proportions, such that the apparent part of the plane covers either the entire width or the entire height of the path resource. Any remaining parts of the path resource that are not covered by the apparent part of the plane are fully transparent. The placement of the apparent part of the plane in the path resource depends on the value of the "adjust" attribute of the <respath> element.

adjust - Applicable only if the value of the "spread" attribute equals 'off'. Optional if applicable.

The adjustment of the apparent part of the plane in the path resource.

The value of the attribute represents an integer between -100 and 100 (inclusive).

The adjustment is applied, either horizontally or vertically, as follows:

* An adjustment of -100 corresponds to the LT placement, where the apparent part of the plane is placed so that its left or top edge matches the left or top edge of the path resource.

* An adjustment of 0 corresponds to the central placement, where the apparent part of the plane is centered vertically or horizontally in the path resource.

* An adjustment of 100 corresponds to the RB placement, where the apparent part of the plane is placed so that its right or bottom edge matches the right or bottom edge of the path resource.

* An adjustment between -100 and 0 (exclusive) corresponds to a proportional placement, where the apparent part of the plane is placed according to the absolute value of the adjustment, starting from the central placement and moving towards the LT placement. For instance, an adjustment of -25 corresponds to a placement where the apparent part of the plane is placed one-quarter of the way from the central placement towards the LT placement.

* An adjustment between 0 and 100 (exclusive) corresponds to a proportional placement, where the apparent part of the plane is placed according to the value of the adjustment, starting from the central placement and moving towards the RB placement. For instance, an adjustment of 75 corresponds to a placement where the apparent part of the plane is placed three-quarters of the way from the central placement towards the RB placement.

The default value is 0.

The value of the attribute is a string in the 'pd' pattern where: 'p' is a substring representing the sign of the adjustment; and 'd' is a substring representing the absolute value of the adjustment. The 'p' substring is either an empty string,

indicating that the value of the adjustment is greater than or equal to zero, or is the '-' (U+002D HYPHEN-MINUS) character. The 'd' substring contains between one and three decimal characters (inclusive). Each decimal character of the 'd' substring is in the range from '0' to '9' (U+0033 DIGIT ZERO to U+0039 DIGIT NINE, inclusive). The 'd' substring does not contain any leading '0' (U+0033 DIGIT ZERO) characters.

As a result, the value of the attribute does not contain any of the following characters: white space characters defined in Section 6, '+' (U+002B PLUS SIGN), and '.' (U+002E FULL STOP).

Examples of accepted values:
adjust='0'
adjust='-100'
adjust='100'
adjust='-25'
adjust='75'

Examples of refused values:
adjust=''
adjust='-0'
adjust='-120'
adjust='150'
adjust='-025'
adjust='075'
adjust='- 25'
adjust='+100'
adjust='-25.5'

color - Optional

The color of the path resource.

The value of the attribute represents the levels of the R, G, and B components of the color. The level of each component is represented using two hexadecimal digits.

As stated in the Frogans slide design concepts (Section 4), the level of each one of the R, G, and B components is in the range from 00 (0) to ff (255).

The default value is '#0000ff'.

The value of the attribute is a string in the '#rrggbb' pattern where: '#' is the U+0013 NUMBER SIGN character; and 'rr', 'gg', and 'bb' are substrings representing respectively the R, G, and B components. Each substring contains two hexadecimal characters.

Each hexadecimal character of each substring is in one of the following ranges: from '0' to '9' (U+0030 DIGIT ZERO to U+0039 DIGIT NINE, inclusive), from 'A' to 'F' (U+0041 LATIN CAPITAL LETTER A to U+0046 LATIN CAPITAL LETTER F, inclusive), from 'a' to 'f' (U+0061 LATIN SMALL LETTER A to U+0066 LATIN SMALL LETTER F, inclusive).

As a result, the value of the attribute does not contain any white space characters defined in Section 6.

Examples of accepted values:

```
color='#0000ff'
color='#004698'
color='#001c46'
color='#ffbf00'
color='#Ffbf00'
color='#FFBF00'
```

Examples of refused values:

```
color=''
color='ffbf00'
color='#'
color='#ffbf'
color='# ffbf00'
color='#ffbf 00'
color='#0ffbf00'
color='#00f'
```

7.12. Adding Layers - <layer> Element

The <layer> element defines a layer to be added to the Frogans slide.

The <layer> element is a child either of the <frogans-fsdl> element (see Section 6.1) or of a <button> element (see Section 7.13). The <layer> element does not contain any children. It does not contain text characters.

As stated in the Frogans slide design concepts (Section 4):

- If the <layer> element is a child of the <frogans-fsdl> element, then the layer is intended to be included directly in the Frogans slide.

- If the <layer> element is a child of the <button> element, then the layer is intended to be used for assembling a button.

- The layer is added on top of any other layers defined previously in the FSDL document.

- The foundation of the layer is a prepared resource, which can be either an image, pixels, draw, path, text, or merge resource (see respectively the <resimage> element in Section 7.2, the <respixels> element in Section 7.3, the <resdraw> element in Section 7.4, the <respath> element in Section 7.5, the <restext> element in Section 7.7, and the <resmerge> element in Section 7.11).

The layer is identified using a component identifier (Section 6) which is determined by the value of the "layerid" attribute of the <layer> element.

The layer can be included in the lead representation, the vignette representation, or both representations of the Frogans slide, depending on the value of the "leapout" attribute of the <layer> element. The values of the other attributes of the <layer> element are applicable to the representations of the Frogans slide in which the layer is included.

The prepared resource used as the foundation of the layer is referred to using a component identifier which is determined by the value of the "resref" attribute of the <layer> element.

The size, in pixels, of the layer on the Frogans slide rendering canvas depends on the size, in pixels, of the prepared resource.

The alignment and position of the layer on the Frogans slide

rendering canvas depend on the values of the "align" and "pos" attributes of the <layer> element.

The following optional transformations are applied successively to the resource:

* flip, depending on the value of the "flip" attribute in the <layer> element

* filter, depending on the value of the "filterref" attribute in the <layer> element

* relief, depending on the value of the "reliefref" attribute in the <layer> element

* blur, depending on the value of the "blur" attribute in the <layer> element

* rotation, depending on the value of the "angle" attribute in the <layer> element

* sharpness, depending on the value of the "sharpness" attribute in the <layer> element

* opacity, depending on the value of the "opacity" attribute in the <layer> element

The transformations are applied to a copy of the prepared resource, which is used for the layer only. As a result, if the prepared resource is to be used as the foundation of other layers defined subsequently in the FSDL document, then the transformations for the layer do not apply to the other layers.

The layer is combined into the Frogans slide rendering canvas in a way that depends on the value of the "combine" attribute of the <layer> element.

A shadow can be added underneath the layer, depending on the value of the "shadowref" attribute of the <layer> element.

If the layer is to be used for assembling a button, then the visibility of the layer in the button depends on the value of the "visible" attribute of the <layer> element.

If the value of the "visible" attribute of a <layer> element is not equal to 'always', then the only possible value of the "combine" attribute is 'clip'.

The reactivity of the layer depends on the value of the "reactivity" attribute of the <layer> element.

The <layer> element has the following attributes:

layerid - Mandatory

The identifier of the layer.

The value of the attribute is a component identifier. The value is compliant with the rules presented in Section 6.

As a result, the value cannot have been utilized previously in the FSDL document as a component identifier.

Examples of accepted values:
layerid='foo1'
layerid='Foo1'
layerid='foo2'
layerid='foo_1'

Examples of refused values:
layerid=''
layerid='foo-1'
layerid='foo/1'
layerid='foo.1'
layerid='foo 1'
layerid='foo1,foo2'

leapout - Mandatory

The way the layer leaps out in the Frogans slide.

The value of the attribute belongs to a set of named values.

The value of the attribute is case-sensitive.

The value can be:

* 'all' indicates that the layer is included in all representations of the Frogans slide, i.e. in both the lead and the vignette representations.

* 'lead' indicates that the layer is included in the lead representation only.

* 'vignette' indicates that the layer is included in the vignette representation only.

If the <layer> element is a child of a <button> element, then the value of the attribute is 'lead'.

resref - Mandatory

The identifier referencing the resource used as the foundation of the layer.

The value of the attribute is a component identifier. The value is compliant with the rules presented in Section 6.

The value of the attribute has been utilized previously in the FSDL document as a component identifier for one of the following resource elements: <resimage> (see Section 7.2), <respixels> (see Section 7.3), <resdraw> (see Section 7.4), <respath> (see Section 7.5), <restext> (see Section 7.7), or <resmerge> (see Section 7.11).

Examples of accepted values:
resref='foo1'
resref='Foo1'
resref='foo2'
resref='foo_1'

Examples of refused values:
resref=''
resref='foo-1'
resref='foo/1'
resref='foo.1'
resref='foo 1'
resref='foo1,foo2'

align - Optional

The way the resource used as the foundation of the layer is aligned relative to the position of the layer on the Frogans slide rendering canvas.

The attribute is based on a string set which represents the possible values of the attribute.

The value of the attribute is case-sensitive.

The value can be:

* 'left-top' indicates that the resource is aligned in a way such that the top-left corner of the resource, as determined before any rotation or blur effect is applied, coincides with the

position of the layer.

* 'center-top' indicates that the resource is aligned in a way such that the point half-way between the top-left and top-right corners of the resource, as determined before any rotation or blur effect is applied, coincides with the position of the layer.

* 'right-top' indicates that the resource is aligned in a way such that the top-right corner of the resource, as determined before any rotation or blur effect is applied, coincides with the position of the layer.

* 'left-middle' indicates that the resource is aligned in a way such that the point half-way between the top-left and bottom-left corners of the resource, as determined before any rotation or blur effect is applied, coincides with the position of the layer.

* 'center-middle' indicates that the resource is aligned in a way such that the vertical and horizontal center of the resource, as determined before any rotation or blur effect is applied, coincides with the position of the layer.

* 'right-middle' indicates that the resource is aligned in a way such that the point half-way between the top-right and bottom-right corners of the resource, as determined before any rotation or blur effect is applied, coincides with the position of the layer.

* 'left-bottom' indicates that the resource is aligned in a way such that the bottom-left corner of the resource, as determined before any rotation or blur effect is applied, coincides with the position of the layer.

* 'center-bottom' indicates that the resource is aligned in a way such that the point half-way between the bottom-left and bottom-right corners of the resource, as determined before any rotation or blur effect is applied, coincides with the position of the layer.

* 'right-bottom' indicates that the resource is aligned in a way such that the bottom-right corner of the resource, as determined before any rotation or blur effect is applied, coincides with the position of the layer.

The default value is 'center-middle'.

During the rendering of the Frogans slide on the Frogans slide rendering canvas, when determining, in pixels, the points half-way between two corners of the resource, as well as the vertical and horizontal center of the resource, non-integer values are rounded down, i.e. towards the top-left corner of the resource.

pos - Mandatory

The position of the layer on the Frogans slide rendering canvas.

The value of the attribute represents the x and y coordinates, in pixels, of the position of the layer on the Frogans slide rendering canvas.

The x coordinate is an integer between -640 and 1,280 (inclusive). The y coordinate is an integer between -480 and 960 (inclusive).

The value of the attribute is a string in the 'px,qy' pattern where: ',' is the U+002C COMMA character; 'p' and 'q' are substrings representing respectively the sign of the x and the y coordinates; and 'x' and 'y' are substrings representing respectively the absolute value of the x and the y coordinates. The 'p' substring is either an empty string, indicating that the value of the x coordinate is greater than or equal to zero, or is the '-' (U+002D HYPHEN-MINUS) character. The 'x' substring contains between one and four decimal characters (inclusive). The 'q' substring is either an empty string, indicating that the value of the y coordinate is greater than or equal to zero, or is the '-' (U+002D HYPHEN-MINUS) character. The 'y' substring contains between one and three decimal characters (inclusive). Each decimal character of the 'x' and 'y' substrings is in the range from '0' to '9' (U+0033 DIGIT ZERO to U+0039 DIGIT NINE, inclusive). The 'x' and 'y' substrings do not contain any leading '0' (U+0033 DIGIT ZERO) characters.

As a result, the value of the attribute does not contain any of the following characters: white space characters defined in Section 6, '+' (U+002B PLUS SIGN), and '.' (U+002E FULL STOP).

Examples of accepted values:
pos='500,300'
pos='1050,300'
pos='-90,300'
pos='500,630'

```
pos='500,-110'

Examples of refused values:
pos=''
pos='-0,300'
pos='2000,300'
pos='-900,300'
pos='500'
pos='500 300'
pos='0500,300'
pos='500, 300'
pos='+500,300'
pos='500.5,300'
```

flip - Optional

The flip effect applied to the resource used as the foundation of the layer.

The attribute is based on a string set which represents the possible values of the attribute.

The value of the attribute is case-sensitive.

The value can be:

* 'none' indicates that the resource is not flipped, i.e. the resource is not modified.

* 'xdir' indicates that the resource is flipped in the x direction (i.e. horizontally), along the vertical axis.

* 'ydir' indicates that the resource is flipped in the y direction (i.e. vertically), along the horizontal axis.

* 'xydir' indicates that the resource is flipped in both the x direction (i.e. horizontally) and the y direction (i.e. vertically), respectively along the vertical and horizontal axes.

The default value is 'none'.

filterref - Optional

The identifier referencing the filter applied to the resource used as the foundation of the layer.

The value of the attribute is either an empty string, or it is a component identifier. In the latter case, the value is compliant with the rules presented in Section 6.

If the value is an empty string, then no filter is applied to the resource.

If the value is a component identifier, then it has been utilized previously in the FSDL document as a component identifier for a <setfilter> element (see Section 7.8).

The default value is '' (the empty string).

```
Examples of accepted values:
filterref=''
filterref='foo1'
filterref='Foo1'
filterref='foo2'
filterref='foo_1'

Examples of refused values:
filterref='foo-1'
filterref='foo/1'
filterref='foo.1'
filterref='foo 1'
filterref='foo1,foo2'
```

For more information on the process to filter a resource, see Appendix B.7.

reliefref - Optional

The identifier referencing the relief effect applied to the resource used as the foundation of the layer.

The value of the attribute is either an empty string, or it is a component identifier. In the latter case, the value is compliant with the rules presented in Section 6.

If the value is an empty string, then no relief effect is applied to the resource.

If the value is a component identifier, then it has been utilized previously in the FSDL document as a component identifier for a <setrelief> element (see Section 7.9).

The default value is '' (the empty string).

```
Examples of accepted values:
reliefref=''
reliefref='foo1'
reliefref='Foo1'
reliefref='foo2'
reliefref='foo_1'

Examples of refused values:
reliefref='foo-1'
reliefref='foo/1'
reliefref='foo.1'
reliefref='foo 1'
reliefref='foo1,foo2'
```

For more information on the process to apply reliefs to a resource, see Appendix B.8.

blur - Optional

The blur effect applied to the resource used as the foundation of the layer.

The value of the attribute represents the x and y radiuses of the blur effect.

The x and y radiuses are integers between 0 and 32 (inclusive).

An x radius of 0 in conjunction with a y radius of 0 corresponds to no blur effect.

The layer includes all pixels, resulting from the blur effect, that extend beyond the edges of the resource.

The default value is '0,0'.

The value of the attribute is a string in the 'xradius,yradius' pattern where: ',' is the U+002C COMMA character; and 'xradius' and 'yradius' are substrings representing respectively the x radius and the y radius of the layer. Each substring contains between one and two decimal characters (inclusive). Each decimal character of the substring is in the range from '0' to '9' (U+0033 DIGIT ZERO to U+0039 DIGIT NINE, inclusive). The substring does not contain any leading '0' (U+0033 DIGIT ZERO) characters.

As a result, the value of the attribute does not contain any of the following characters: white space characters defined in Section 6, '+' (U+002B PLUS SIGN), '-' (U+002D HYPHEN-MINUS), and '.' (U+002E FULL STOP).

```
Examples of accepted values:
blur='0,0'
blur='32,32'
blur='5,0'
blur='0,5'
blur='5,32'

Examples of refused values:
blur=''
blur='5,44'
blur='5'
blur='5 32'
blur='05,32'
blur='5, 32'
blur='+5,32'
blur='5,-32'
blur='5.5,32'
```

For more information on the process to blur a resource, see Appendix B.9.

angle - Optional

The angle of rotation, in degrees, applied to the resource used as the foundation of the layer.

The value of the attribute represents an integer between -180 and 180 (inclusive).

The rotation is applied as follows:

* An angle between -180 and -1 (inclusive) corresponds to counterclockwise rotation.

* An angle of 0 corresponds to no rotation, i.e. the resource is not modified.

* An angle between 1 and 180 (inclusive) corresponds to clockwise rotation.

The default value is 0.

The value of the attribute is a string in the 'pn' pattern where: 'p' is a substring representing the sign of the angle; and 'n' is a substring representing the absolute value of the angle. The 'p' substring is either an empty string, indicating that the value of the angle is greater than or equal to zero, or is the '-' (U+002D HYPHEN-MINUS) character. The 'n' substring contains between one

and three decimal characters (inclusive). Each decimal character of the 'n' substring is in the range from '0' to '9' (U+0033 DIGIT ZERO to U+0039 DIGIT NINE, inclusive). The 'n' substring does not contain any leading '0' (U+0033 DIGIT ZERO) characters.

As a result, the value of the attribute does not contain any of the following characters: white space characters defined in Section 6, '+' (U+002B PLUS SIGN), and '.' (U+002E FULL STOP).

Examples of accepted values:
angle='0'
angle='-180'
angle='180'
angle='-45'
angle='90'

Examples of refused values:
angle=''
angle='-0'
angle='-240'
angle='360'
angle='-045'
angle='090'
angle='- 45'
angle='+90'
angle='-45.5'

For more information on the process to rotate a resource, see Appendix B.10.

sharpness - Optional

The sharpness effect applied to the resource used as the foundation of the layer.

The value of the attribute represents an integer between 0 and 8 (inclusive).

A sharpness of 0 corresponds to no sharpness effect.

The default value is 0.

The value of the attribute is a string containing one decimal character. The decimal character of the string is in the range from '0' to '8' (U+0030 DIGIT ZERO to U+0038 DIGIT EIGHT, inclusive).

As a result, the value of the attribute does not contain any of the following characters: white space characters defined in Section 6, '+' (U+002B PLUS SIGN), '-' (U+002D HYPHEN-MINUS), and '.' (U+002E FULL STOP).

Examples of accepted values:
sharpness='0'
sharpness='8'
sharpness='5'

Examples of refused values:
sharpness=''
sharpness='9'
sharpness='+5'
sharpness='-5'
sharpness='5.5'

For more information on the process to sharpen a resource, see Appendix B.11.

opacity - Optional

The opacity effect applied to the resource used as the foundation of the layer.

The value of the attribute represents an integer between 0 and 100 (inclusive).

The opacity effect is applied as follows:

* An opacity of 0 corresponds to an opacity effect where the resource becomes fully transparent.

* An opacity of 100 corresponds to no opacity effect, i.e. the resource is not modified.

* An opacity between 0 and 100 (exclusive) corresponds to an opacity effect where the A component (transparency) of each visible pixel in the resource is reduced in proportion to the opacity. For instance, an opacity of 75 corresponds to reducing the A component (transparency) of each visible pixel in the resource to three-quarters of the its value.

As a result, the opacity effect does not increase the A component (transparency) of any pixel in the resource.

The default value is '100'.

The value of the attribute is a string containing between one and three decimal characters (inclusive). Each decimal character of the string is in the range from '0' to '9' (U+0030 DIGIT ZERO to U+0039 DIGIT NINE, inclusive). The string does not contain any leading '0' (U+0033 DIGIT ZERO) characters.

As a result, the value of the attribute does not contain any of the following characters: white space characters defined in Section 6, '+' (U+002B PLUS SIGN), '-' (U+002D HYPHEN-MINUS), and '.' (U+002E FULL STOP).

Examples of accepted values:
opacity='100'
opacity='0'
opacity='75'

Examples of refused values:
opacity=''
opacity='-25'
opacity='150'
opacity='075'
opacity='+75'
opacity='75.5'

combine - Mandatory

The way the layer is combined into the Frogans slide rendering canvas.

The attribute is based on a string set which represents the possible values of the attribute.

The value of the attribute is case-sensitive.

The value can be:

* 'add' indicates that the layer is added into the Frogans slide rendering canvas. This way of combining can modify both the A component (transparency) and the R, G, and B components (color) of pixels of the Frogans slide rendering canvas.

* 'clip' indicates that the layer is clipped into the Frogans slide rendering canvas. This way of combining can modify the R, G, and B components (color) of pixels in the Frogans slide rendering canvas. It cannot modify the transparency (A component) of any pixels of the Frogans slide rendering canvas.

* 'cutout' indicates that the layer is cut out from the Frogans slide rendering canvas. This way of combining can modify the A component (transparency) of pixels of the Frogans slide rendering canvas. It cannot modify the R, G, and B components (color) of any pixels of the Frogans slide rendering canvas.

* 'inter' indicates that the layer is intersected with the Frogans slide rendering canvas. This way of combining can modify the A component (transparency) of pixels of the Frogans slide rendering canvas. It cannot modify the R, G, and B components (color) of any pixels of the Frogans slide rendering canvas.

As a result, if the layer is the first layer defined in the FSDL document and the value is not equal to 'add', then the layer is not visible in the Frogans slide.

For more information on the process to combine layers, see Appendix B.12.

shadowref - Optional

The identifier referencing the shadow added underneath the layer.

The value of the attribute is either an empty string, or it is a component identifier. In the latter case, the value is compliant with the rules presented in Section 6.

If the value is an empty string, then no shadow is added underneath the layer.

If the value is a component identifier, then it has been utilized previously in the FSDL document as a component identifier for a <setshadow> element (see Section 7.10).

The default value is '' (the empty string).

```
Examples of accepted values:
shadowref=''
shadowref='foo1'
shadowref='Foo1'
shadowref='foo2'
shadowref='foo_1'

Examples of refused values:
shadowref='foo-1'
```

```
shadowref='foo/1'
shadowref='foo.1'
shadowref='foo 1'
shadowref='foo1,foo2'
```

For more information on the process to apply shadows underneath a layer, see Appendix B.13.

visible - Applicable only if the <layer> element is a child element of a <button> element. Mandatory if applicable.

The visibility of the layer of the button.

The attribute is based on a string set which represents the possible values of the attribute.

The value of the attribute is case-sensitive.

The value can be:

* 'always' indicates that the layer of the button is always visible, i.e. whether or not the button is selected.

* 'not-selected' indicates that the layer of the button is visible only if the button is not selected.

* 'selected' indicates that the layer of the button is visible only if the button is selected.

reactivity - Optional

The reactivity of the layer of the button.

The default value is '#7f'.

The value of the attribute is a string in the '#aa' pattern where: '#' is the U+0013 NUMBER SIGN character; and 'aa' is a substring representing the A component. Each substring contains two hexadecimal characters. Each hexadecimal character of each substring is in one of the following ranges: from '0' to '9' (U+0030 DIGIT ZERO to U+0039 DIGIT NINE, inclusive), from 'A' to 'F' (U+0041 LATIN CAPITAL LETTER A to U+0046 LATIN CAPITAL LETTER F, inclusive), from 'a' to 'f' (U+0061 LATIN SMALL LETTER A to U+0066 LATIN SMALL LETTER F, inclusive).

As a result, the value of the attribute does not contain any white space characters defined in Section 6.

```
Examples of accepted values:
reactivity='#ff'
reactivity='#00'
reactivity='#bf'
reactivity='#Bf'
reactivity='#BF'

Examples of refused values:
reactivity=''
reactivity='bf'
reactivity='#'
reactivity='#b'
reactivity='# bf'
reactivity='#b f'
reactivity='#0bf'
```