



☐ need examples  
☑ need Medical review  
☐ DTD Schema Tools tips  
once node names decided

# Extensible 3D (X3D) Part 1: Architecture and base components

## 42 Annotation component



### 42.1 Introduction

#### 42.1.1 Name

The name of this component is "Annotation". This name shall be used when referring to this component in the COMPONENT statement (see 7.2.5.4 Component statement).

#### 42.1.2 Overview

This clause describes the Annotation component of this part of ISO/IEC 19775. This component specifies how to include additional information as part of an X3D file that is beyond basic metadata, and some or all of which may be displayed externally to the scene. Table 42.1 provides links to the major topics in this clause.

**Table 42.1 – Topics**

- 42.1 Introduction
  - 42.1.1 Name
  - 42.1.2 Overview
- 42.2 Concepts
  - 42.2.1 Overview
  - 42.2.2 Representing annotations
    - 42.2.2.1 Parts of an annotation
    - 42.2.2.2 Coordinate system
- 42.3 Abstract types
  - 42.3.1 X3DAnnotationNode
- 42.4 Node reference
  - 42.4.1 AnnotationLaver

- [42.4.3 GroupAnnotation](#)
  - [42.4.4 IconAnnotation](#)
  - [42.4.5 TextAnnotation](#)
  - [42.4.6 URLAnnotation](#)
- [42.5 Support levels](#)

- [Table 42.1 — Topics](#)
- [Table 42.2 — Display policies](#)
- [Table 42.3 — Layout policies](#)
- [Table 42.4 — Annotation component support levels](#)

## ● 42.2 Concepts

### 42.2.1 Overview

Annotations occupy the information space between metadata provided by the core specification and the rendered visuals presented by the rest of the X3D specification. The goal is to provide the ability to include or give reference to large amounts of information that may be available in scene or, optionally at the browser implementer's choice, externally to the scene. Another term for annotations is "labelling" which implies purely in-scene information, where this specification seeks to provide both in-scene and out-of-scene information display capabilities.

### 42.2.2 Representing annotations

#### 42.2.2.1 Parts of an annotation

An annotation is comprised of three parts:

1. the object that is being annotated,
2. the information to be associated with that object, and
3. a visual connection between the two, such as a headline.

The headline and information may be optionally presented by the browser.

EXAMPLE A hide-away panel may be used to display annotations that are available for currently visible objects.

In addition, the user is provided with an ability to describe the preferred policy for the browser to show annotations that are not currently visible, such as when they are behind the current viewpoint.

#### 42.2.2.2 Coordinate system

The object that is being annotated <sup>has</sup> will have the AnnotationTarget node

associated with the containing grouping node. This node is not renderable, but provides a point in the local coordinate system that forms the end point of the annotation's leadline connecting that object.

Annotations themselves are not part of the renderable scene graph. They may, optionally, exist anywhere and are not affected by the parent transformation hierarchy, such as Switch or LOD nodes. The AnnotationTarget node is effected by parent transformation hierarchy changes, including those that disable parts of the visible scene graph. When the parent hierarchy hides the renderable parts, the target and any associated leadlines are also hidden.

### 42.3 Abstract types

#### 42.3.1 X3DAnnotationNode

```

X3DAnnotationNode : X3DChildNode {
  SFString [in,out] annotationGroupID ""
  SFString [in,out] displayPolicy "NEVER" ["POINTER_OVER", "POINTER_ACTIVATE",
                                           "ALWAYS", "WHEN_VISIBLE", "NEVER"]
  SFFloat [in,out] enabled TRUE
  SFNode [in,out] metadata NULL [X3DMetadataObject]

```

*[proposed] put url here.*

*others?*

This abstract node type is the base type for all node types that contain annotation information. The abstract type provides the facility to control the policy of when the annotation should be displayed to the user through the *displayPolicy* field. Additionally, a grouping label can be assigned to each annotation through the use of the *annotationGroupID* field.

Large files with many annotations can become unwieldy to view. Two options are available to control how and when annotations are to be made visible. The first option is to create a global filter list through the use of the *annotationGroupID* field. This is a text string that can be used to group sets of similar annotations together. It is recommended that plain, readable text be used for this field as a browser may present the values of these fields in a list of filter options. The second option controls when an individual annotation will show based on user actions. These provide basic shortcuts without the user needing to use combinations of touch sensors and scripting. Available basic behaviours are in Table 42.2:

*for consistent presentation*

*is displayed*

*choices rather for such*

*described*

**Table 42.2 – Display policies**

POINTER_OVER	Show the annotation when the pointing device is over the targets that reference this node.
POINTER_ACTIVATE	Show this annotation when the pointing device has clicked on the target of this node. It will remain active until replaced by another annotation.
WHEN_VISIBLE	Show this annotation when it is visible to users <i>field of view</i> .

ALWAYS	Always show this annotation.
NEVER	Never show this annotation.

The *enabled* field determines if the annotation is displayed. If *enabled* is TRUE, the annotation is displayed as described. If *enabled* is FALSE, no action is performed.

*If a node is associated with the associated Annotation node instance, its content is reloaded whenever enabled is changed from FALSE to TRUE.*

### 42.4 Node reference

#### 42.4.1 AnnotationLayer

```
AnnotationLayer : X3DLayerNode {
  SFBool [in,out] isPickable TRUE
  MFString [in,out] layoutPolicy "" ["circular", "edges", ...]
  SFNode [in,out] metadata NULL [X3DMetadataObject]
  MFString [in,out] shownGroupID []
  SFNode [in,out] viewport NULL [X3DViewportNode]
}
```

*node dependent on layering component add there*

*supports the ability to define ~~edges~~ consistent*

The AnnotationLayer node is custom layer version that provides automated layout of currently visible annotations and display of them within the currently running scene.

*When active, AnnotationLayer*

The layer shows annotations directly in the scene and lays them out around the target object's reference point according to one of the pre-defined layout policies that are defined in the *layoutPolicy* field. This field contains a list of policies in priority order based on what the browser implementation supports. Browsers may also define implementation-specific policies in addition to the required policies. The reference point is projected into the layer's space and used as the location to base the annotations around. The defined policies are specified in Table 42.3:

*[check required]*

*Additional functionality is defined in Layer component.*

Table 42.3 – Layout policies

CIRCULAR	Show the annotation when the pointing device is over the targets that reference this node. <i>Annotation information is presented in a circular fashion around the target to cover.</i>
<del>DISPLAY EDGES</del>	Show this annotation when the pointing device has clicked on the target of this node. It will remain active until replaced by another annotation.

*is Pickable refers to functionality in picking component, where user can select*

*Edges are defined around each of the relevant annotations being presented.*

#### 42.4.2 AnnotationTarget

```
AnnotationTarget : X3DChildNode {
  MFNode [in,out] annotations [] [X3DAnnotationNode]
  SFNode [in,out] leadLineStyle NULL [X3DLinePropertiesNode]
  SFNode [in,out] marker NULL [X3DShapeNode]
  SFNode [in,out] metadata NULL [X3DMetadataObject]
  SFVec3f [in,out] referencePoint 0, 0, 0 (-∞, ∞)
```

The AnnotationTarget node specifies the target with which annotations are

associated. A target applies to all siblings of the parent grouping node. If the parent or target has more than one parent transformation hierarchy, each shall be rendered individually, including leader lines and markers to each visual object.

Each target may have zero or more annotation nodes associated with it as specified by the *annotations* field. All nodes referenced in this field are effected <sup>presented</sup> at the position of this node's parent group in the world coordinates.

A grouping node may have more than one AnnotationTarget specified, each with different sets of annotations.

Annotations are visually connected to the target through the use of a lead line if a node is defined for the *leadLineStyle* field. If no style is defined, a lead line is not shown and the annotation is shown with no connecting line. When a LineProperties node is provided and line ends are defined, the source end <sup>is</sup> the target and the destination end <sup>is</sup> the annotation. <sup>location</sup> <sup>displayed</sup>

### 42.4.3 GroupAnnotation <sup>← better name. ModelAnnotation</sup>

```

GroupAnnotation : X3DGroupingNode, X3DAnnotationNode {
  MFNode [in]      addChildren      [X3DChildNode]
  MFNode [in]      removeChildren   [X3DChildNode]
  SFString [in,out] annotationGroupID ""
  MFNode [in,out] children          []      [X3DChildNode]
  SFString [in,out] displayPolicy    "NEVER" ["POINTER_OVER", "POINTER_ACTIVATE",
                                                "ALWAYS", "WHEN_VISIBLE", "NEVER"]

  SFBool [in,out] enabled            TRUE
  SFNode [in,out] metadata           NULL   [X3DMetadataObject]
  SFVec3f []      bboxCenter          0 0 0  (-∞,∞)
  SFVec3f []      bboxSize            -1 -1 -1 [0,∞) or -1 -1 -1
}

```

The GroupAnnotation node specifies annotation that is specified in the form of an X3D grouping node.

The annotation to be displayed is specified in the *children* field. The coordinate system established for this group is one in which the origin is at the end of the offset sequence with the XY-plane parallel with the screen plane. From this coordinate system the nodes in the *children* field may apply further transformations. See 4.3.5, Transformation hierarchy, and 4.3.6, Standard units and coordinate system, for a description of coordinate systems and transformations.

10.2.1, Grouping and children node types, provides a description of the *children*, *addChildren*, and *removeChildren* fields.

The *bboxCenter* and *bboxSize* fields specify a bounding box that encloses the children of the GroupAnnotation node. This is a hint that may be used for optimization purposes. The results are undefined if the specified bounding box is smaller than the actual bounding box of the children at any time. A default *bboxSize* value, (-1, -1, -1), implies that the bounding box is not specified and, if needed, shall be calculated by the browser. The bounding

box shall be large enough at all times to enclose the union of the group's children's bounding boxes; it shall not include any transformations performed by the group itself (i.e., the bounding box is defined in the local coordinate system of the children). The results are undefined if the specified bounding box is smaller than the true bounding box of the group. A description of the *bboxCenter* and *bboxSize* fields is provided in 10.2.2 Bounding boxes.

### 42.4.4 <sup>Image</sup> IconAnnotation

*Image Annotation more general  
Icon implies size/shape restrictions*

```
IconAnnotation : X3DAnnotationNode, X3DURLObject {
  SFString [in,out] annotationGroupID ""
  SFString [in,out] displayPolicy "NEVER" ["POINTER_OVER", "POINTER_ACTIVATE",
                                           "ALWAYS", "WHEN_VISIBLE", "NEVER"]
  SFBool [in,out] enabled TRUE
  SFNode [in,out] metadata NULL [X3DMetadataObject]
  MFString [in,out] url []
}
```

The IconAnnotation node specifies annotation that is iconic in form using an image specified by the URL field.

The icon to be displayed is read from location specified by the *url* field. When the *url* field contains no satisfiable values, the browser implementation shall substitute a default icon in its place. Browsers shall support the JPEG (see 2.[JPEG]) and PNG (see ISO/IEC 15948) image file formats. Browsers may support other image file formats. Details on the *url* field can be found in 9.2.1 URLs.

### 42.4.5 TextAnnotation

```
TextAnnotation : X3DAnnotationNode {
  SFString [in,out] annotationGroupID ""
  SFString [in,out] contentType "text/plain"
  SFString [in,out] displayPolicy "NEVER" ["POINTER_OVER", "POINTER_ACTIVATE",
                                           "ALWAYS", "WHEN_VISIBLE", "NEVER"]
  SFBool [in,out] enabled TRUE
  SFNode [in,out] metadata NULL [X3DMetadataObject]
  SFString [in,out] text ""
}
```

*proposed*

*if text url is provided, it may override text field defined.*  
The TextAnnotation node specifies an annotation that contains ~~unformatted~~ formatted text. The text may be one of several formats based on the defined MIME type in the *contentType* field. All browsers shall support the default plain text content type, and may support other content types (e.g., HTML).

*[Similar effect as provide vs local url]*

*unstructured or*

### 42.4.6 URLAnnotation

```
URLAnnotation : X3DAnnotationNode {
  SFString [in,out] annotationGroupID ""
  SFString [in,out] displayPolicy "NEVER" ["POINTER_OVER", "POINTER_ACTIVATE",
                                           "ALWAYS", "WHEN_VISIBLE", "NEVER"]
  SFBool [in,out] enabled TRUE
  SFNode [in,out] metadata NULL [X3DMetadataObject]
  MFString [in,out] url []
}
```

*proposed*  
*remove this node, integrate into X3D Annotation Node*

The URLAnnotation node specifies an annotation that defines its content in

another file.

The location of the other file is defined by the *url* field. This annotation is not required to be immediately loaded. A browser may choose to load the URL or just display the URL for the user to select and load externally.

*The file referred to by the url may contain X3D geometry usable as a Group Annotation child, text information usable as a Text Annotation string, or an image file usable as an Image Annotation. Security precaution: no recursive loading.*

● **42.5 Support levels**

The Volume Rendering component provides two levels of support as specified in Table 42.4.

**Table 42.4 – Annotation component support levels**

Level	Prerequisites	Nodes/Features	Support
1	Core 1 Grouping 1 Shape 1 Rendering 1 Networking 1		
		X3DAnnotationNode	n/a
		AnnotationTarget	All fields fully supported.
		IconAnnotation	All fields fully supported.
		TextAnnotation	All fields fully supported.
	<i>problem: what if Group/Model Annotation?</i>	URLAnnotation	All fields fully supported.
2	Core 1 Grouping 1 Shape 1 Rendering 1 Networking 1, Layering 1		
		All Level 1 nodes	All fields fully supported.
		AnnotationLayer	All fields fully supported.
		GroupAnnotation	All fields fully supported.

