

First edition
2006-06-15

AMENDMENT 1
2008-03-01

Corrected version
2008-04-15

**Information technology — Coding of
audio-visual objects —**

Part 20:

**Lightweight Application Scene
Representation (LAsER) and Simple
Aggregation Format (SAF)**

AMENDMENT 1: LAsER extensions

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC 14496-20:2006/Amd.1:2008
<https://standards.iteh.ai/catalog/standards/sist/8dbba16d-bc1d-4309-9f33-b4f0d9e2544c/iso-iec-14496-20-2006-amd-1-2008>
*Technologies de l'information — Codage des objets audiovisuels —
Partie 20: Représentation de scène d'application allégée (LAsER) et
format d'agrégation simple (SAF)*

AMENDEMENT 1: Extensions LAsER

Reference number
ISO/IEC 14496-20:2006/Amd.1:2008(E)



PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[ISO/IEC 14496-20:2006/Amd 1:2008](https://standards.iteh.ai/catalog/standards/sist/8dbba16d-bc1d-4309-9f33-b4f0d9e2544e/iso-iec-14496-20-2006-amd-1-2008)

<https://standards.iteh.ai/catalog/standards/sist/8dbba16d-bc1d-4309-9f33-b4f0d9e2544e/iso-iec-14496-20-2006-amd-1-2008>



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2008

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Amendment 1 to ISO/IEC 14496-20:2006 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This corrected version of ISO/IEC 14496-20:2006/Amd.1:2008 incorporates the following corrections: a change to the title of the amendment and the provision of updated electronic attachments relevant to LASER.

<https://standards.iteh.ai/catalog/standards/sist/8dbba16d-bc1d-4309-9f33-b4f0d9e2544e/iso-iec-14496-20-2006-amd-1-2008>

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[ISO/IEC 14496-20:2006/Amd 1:2008](https://standards.iteh.ai/catalog/standards/sist/8dbba16d-bc1d-4309-9f33-b4f0d9e2544e/iso-iec-14496-20-2006-amd-1-2008)

<https://standards.iteh.ai/catalog/standards/sist/8dbba16d-bc1d-4309-9f33-b4f0d9e2544e/iso-iec-14496-20-2006-amd-1-2008>

Information technology — Coding of audio-visual objects —

Part 20:

Lightweight Application Scene Representation (LAsER) and Simple Aggregation Format (SAF)

AMENDMENT 1: LAsER extensions

Replace all occurrences of the word “browser” with “LAsER engine”.

In Clause 2, add the following references:

ISO/IEC 9899:1990, *Information technology — Programming Languages C*

ISO/IEC 14882, *Programming languages — C++*

ISO/IEC 16262:2002, *Information technology — ECMA Script language specification*

IETF BCP 13, *RFC 4288 on Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures*

IETF RFC 3023, XML Media Types, M. Murata, S. St Laurent, D. Kohn, January 2001, <http://www.ietf.org/rfc/rfc3023.txt>

IETF RFC 3986, Uniform Resource Identifiers (URI): Generic Syntax, T. Berners-Lee, R. Fielding, L. Masinter, January 2005, <http://www.ietf.org/rfc/rfc3986.txt>

Add the following definition of waiting tree to clause 3:

waiting tree

separate tree defined in addition to the scene tree

NOTE The compositor and renderer have no knowledge of the waiting tree, thus objects in the waiting tree are neither composited nor rendered.

Add the following subclause to Clause 5:

5.2 LAsER Systems Decoder Model

5.2.1 Introduction

The purpose of the LAsER Systems decoder model is to provide an abstract view of the behaviour of the terminal complying with this International Standard. It may be used by the sender to predict how the receiving terminal will behave in terms of buffer management and synchronization when decoding data received in the

form of elementary streams. The LAsER systems decoder model includes a timing model and a buffer model. The LAsER systems decoder model specifies:

1. the conceptual interface for accessing data streams (Delivery Layer),
2. decoding buffers for coded data for each elementary stream,
3. the behavior of elementary stream decoders,
4. composition memory for decoded data from each decoder, and
5. the output behavior of composition memory towards the compositor.

Each elementary stream is attached to one single decoding buffer.

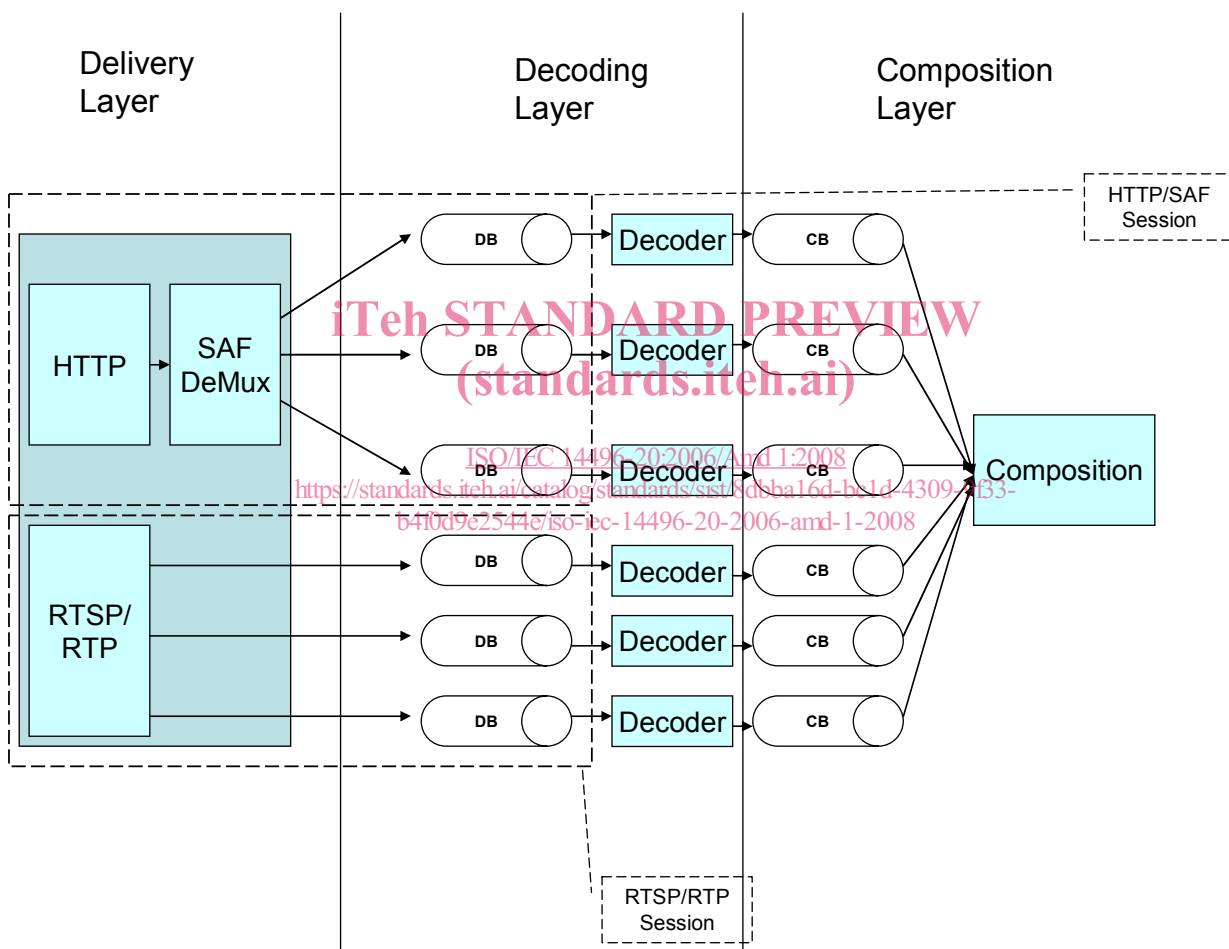


Figure AMD1.1 – LAsER Systems Decoder Model

The definition in ISO/IEC 14496-1 of Access Unit, Decoding Buffer(DB), elementary stream (ES), Decoder (CU) and Composition Unit apply.

5.2.2 Decoder Model

The decoder model as specified in ISO/IEC 14496-1:2004 subclause 7.4.1 applies.

5.2.3 Decoding Buffer

The needed decoding buffer size is known by the sending terminal and conveyed to the receiving terminal as specified in 7.6. The size of the decoding buffer is measured in bytes. The decoding buffer is filled at the rate given by the maximum bit rate for this elementary stream while data is available and with a zero rate otherwise. The maximum bit rate is conveyed by the sending terminal as a part of the decoder configuration information during the set up phase for each elementary stream (see 7.6).

5.2.4 Decoder model with grouped streams

This decoder model may be enhanced when used for group of multiple elementary streams.

In such case, only one composition buffer for the group of streams is used for composition.

When such streams are grouped, and when the setup of multiple decoding chains are available, it is possible, although not mandatory, not to decode all streams at a time.

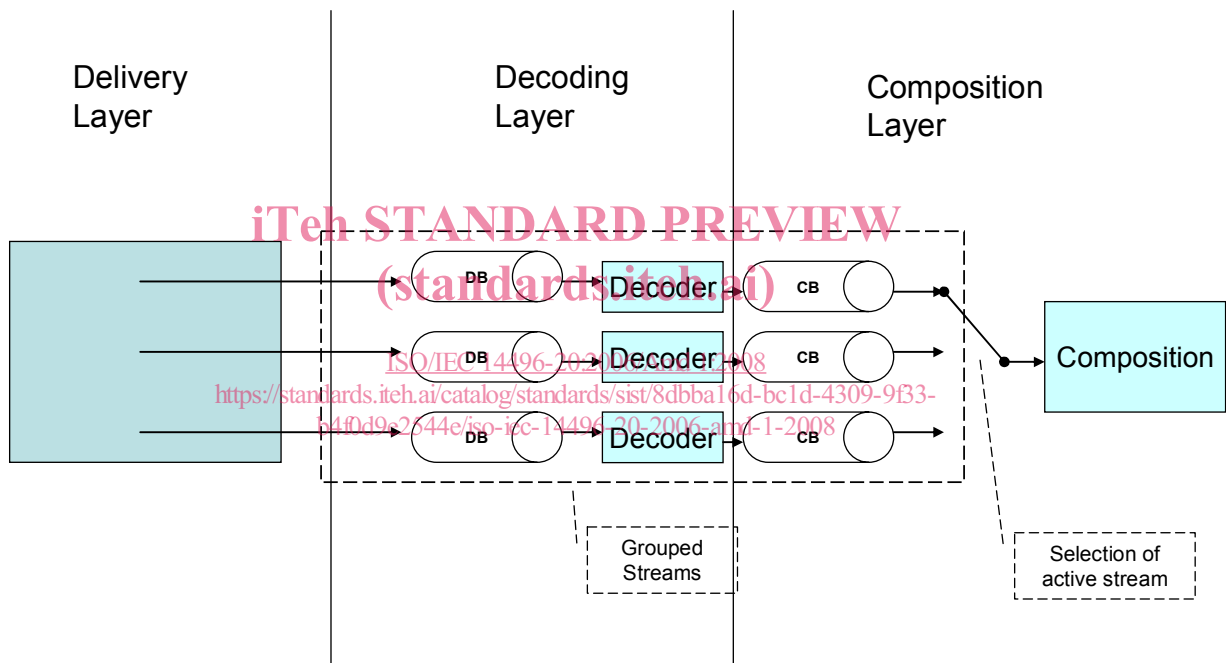


Figure AMD1.2 – Stream grouping with specified System Decoder Model (multiple decoders)

It is indeed expected that multiple decoders may not be available in lightweight terminals or that some delivery scenarios do not allow for having all streams available at the same time (e.g. in broadcast scenarios, the delivery layer could only tune in to one of the streams). The usage of new information about this grouping enables a smart usage of buffers and decoders.

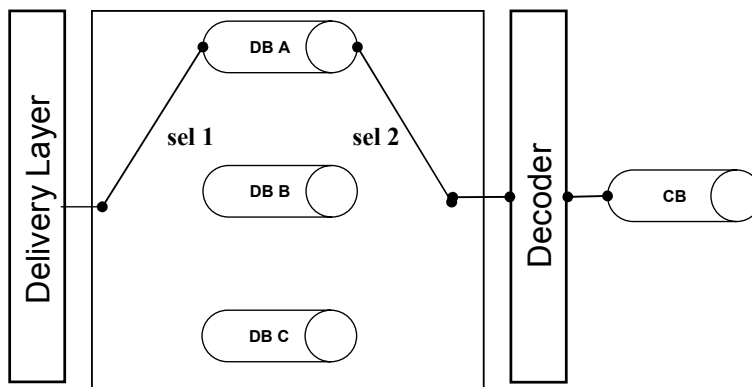


Figure AMD1.3 – Broadcast example of streams grouping, showing a potential optimization using a single decoder.

When only a subset of the group of streams can be accessed at a time (e.g. broadcast scenario depicted above), the selection of the active stream corresponds to a request for the corresponding streams. Nevertheless, the buffer model for stream grouping does not assume immediate reception of data after such request and therefore the active decoding buffer may continue to be used by the decoder up to the moment at which data is available for the newly available stream. In this case the decoding buffer associated to the newly connected stream can be associated with the decoder. At this point the terminal may discard any remaining access units in the previous decoding buffer.

Add the following text to subclause 6.2.1 Scene Tree, at the end:

ISO/IEC 14496-20:2006/Amd 1:2008

The API defined in Appendix A of [2] with IDL definitions in Appendix B of the same document can be used to access the LAsER scene tree from programming languages such as ECMA-Script [ISO/IEC 16262], Java [5], C [ISO/IEC 9899:1990] or C++ [ISO/IEC 14882:2002].

Replace subclause 6.5 with:

6.5. Events

6.5.1. Purposes of events

As in SVG, LAsER defines events following the XML Event specification [6]. The events defined in LAsER relate to the management of the network session and decoding chains (including decoding buffers). The events defined in the following subclauses can be used by elements in the scene such as script elements being associated, through the listener element, in order to respond to such events.

Note : For instance, in a progressive download scenario, the "buffering" event could be listened by a script in order to trigger a text indicating that content will be played shortly.

```

...
<ev:listener handler="#myscript" event="LAsERBuffering">
<script id="myscript">
  <lsr:Replace ref="#text" attributeName="visibility" value="visible">
</script>
<text id="text" visibility="hidden">Content is being buffered</text>
...

```


In the previous example, the LAsERBuffering event is being listened to by a script "myscript". When the event is launched by the browser, the visibility attribute of the text element is set to "visible".

These events are launched by the LAsER browser either at the "Network" Layer of the browser in which case, the scope of these events is the session or at the decoding chain level, in which case the events are at the stream level.

NOTE a session is identified by a unique url and streams are identified by a streamID.

6.5.2. Events imported from SVG Tiny

The list of supported events with their properties is given in Table AMD1.1.

Table AMD1.2 – List of supported events from SVGT1.2

Event name	Namespace	Description	Bubble	Canc.
"focusin" or "DOMFocusIn"	http://www.w3.org/2001/xml-events	As defined in subclause 13.2 of [2].	Yes	No
"focusout" or "DOMFocusOut"	http://www.w3.org/2001/xml-events	As defined in subclause 13.2 of [2].	Yes	No
"activate" or "DOMActivate"	http://www.w3.org/2001/xml-events	As defined in subclause 13.2 of [2].	Yes	Yes
"click"	http://www.w3.org/2001/xml-events	As defined in subclause 13.2 of [2].	Yes	Yes
"mousedown"	http://www.w3.org/2001/xml-events	As defined in subclause 13.2 of [2].	Yes	Yes
"mouseup"	http://www.w3.org/2001/xml-events	As defined in subclause 13.2 of [2].	Yes	Yes
"mouseover"	http://www.w3.org/2001/xml-events	As defined in subclause 13.2 of [2].	Yes	Yes
"mouseout"	http://www.w3.org/2001/xml-events	As defined in subclause 13.2 of [2].	Yes	Yes
"mousemove"	http://www.w3.org/2001/xml-events	As defined in subclause 13.2 of [2].	Yes	No
"load" (or deprecated "SVGLoad")	http://www.w3.org/2001/xml-events	As defined in subclause 13.2 of [2].	No	No
"resize" (or deprecated "SVGResize")	http://www.w3.org/2001/xml-events	As defined in subclause 13.2 of [2].	Yes	No
"scroll" (or deprecated "SVGScroll")	http://www.w3.org/2001/xml-events	As defined in subclause 13.2 of [2].	Yes	No
"zoom" (or deprecated "SVGZoom")	http://www.w3.org/2001/xml-events	As defined in subclause 13.2 of [2].	Yes	No
"beginEvent"	http://www.w3.org/2001/xml-events	As defined in subclause 13.2 of [2].	Yes	No
"endEvent"	http://www.w3.org/2001/xml-events	As defined in subclause 13.2 of [2].	Yes	No
"repeatEvent"	http://www.w3.org/2001/xml-events	As defined in subclause 13.2 of [2].	Yes	No
"keyup"	http://www.w3.org/2001/xml-events	As defined in subclause 13.2 of [2].	No	No
"keydown"	http://www.w3.org/2001/xml-events	As defined in subclause 13.2 of [2].	No	No
"textInput"	http://www.w3.org/2001/xml-events	As defined in subclause 13.2 of [2].	No	No
"mouseWheel"	http://www.w3.org/2001/xml-events	As defined in subclause 13.2 of [2].	No	No
"timer"	http://www.w3.org/2001/xml-events	As defined in subclause 13.2 of [2].	No	No
"preload"	http://www.w3.org/2000/svg	As defined in subclause 13.2 of [2].	No	No
"loadProgress"	http://www.w3.org/2000/svg	As defined in subclause 13.2 of [2].	No	No
"postLoad"	http://www.w3.org/2000/svg	As defined in subclause 13.2 of [2].	No	No
"connectionConnected"	http://www.w3.org/2000/svg	As defined in subclause 13.2 of [2].	No	No
"connectionClosed"	http://www.w3.org/2000/svg	As defined in subclause 13.2 of [2].	No	No
"connectionError"	http://www.w3.org/2000/svg	As defined in subclause 13.2 of [2].	No	No
"connectionDataSent"	http://www.w3.org/2000/svg	As defined in subclause 13.2 of [2].	No	No
"connectionDataReceived"	http://www.w3.org/2000/svg	As defined in subclause 13.2 of [2].	No	No

6.5.3. Pseudo-events

Pseudo-events are shortcuts created by combinations of other events. Their definition follows:

Table AMD1.3 – List of pseudo-events

Event name	Namespace	Description	Bubble	Canc.
“accessKey(keyCode)”	http://www.w3.org/2000/svg	The key keyCode has been pressed, as defined in subclause 16.2.7 of [2]. This pseudo-event is triggered immediately by a listener on keydown placed on the document node.	No	No
“longAccessKey(keyCode)”	urn:mpeg:mpeg4:laser:2005	This pseudo-event is a combination of a listener on keydown and a listener on keyup placed on the document node; this pseudo-event is triggered if after a system-defined time A after the keydown event, no keyup event has happened.	No	No
“repeatKey(keyCode)”	urn:mpeg:mpeg4:laser:2005	This pseudo-event is a combination of a listener on keydown and a listener on keyup placed on the document node; this pseudo-event is triggered after a system-defined time B after the keydown event, repeatedly every system-defined period C, until a keyup event happens.	No	No
“shortAccessKey(keyCode)”	urn:mpeg:mpeg4:laser:2005	This pseudo-event is a combination of a listener on keydown and a listener on keyup placed on the document node; this pseudo-event is triggered if a keyup event happens after a time shorter than A after the keydown event. This is exclusive of longAccessKey(k).	No	No

iTeh STANDARD PREVIEW
(standards.iteh.ai)

6.5.4. LAsER Events

LAsER defines the following events: [ISO/IEC 14496-20:2006/Amd 1:2008](https://standards.iteh.ai/catalog/standards/sist/8dbba16d-bc1d-4309-9f33-b4fd9e2544e/iso-iec-14496-20-2006-amd-1-2008)
<https://standards.iteh.ai/catalog/standards/sist/8dbba16d-bc1d-4309-9f33-b4fd9e2544e/iso-iec-14496-20-2006-amd-1-2008>

Table AMD1.4 – List of LAsER events

Event name	Namespace	Description	Bubble	Canc.
“pause”	urn:mpeg:mpeg4:laser:2005	Freezes the clock of the timed object they are sent to, and have no effect on non timed objects.	No	No
“play”	urn:mpeg:mpeg4:laser:2005	Starts or resumes the clock of the timed object they are sent to, and have no effect on non timed objects.	No	No
“pausedEvent”	urn:mpeg:mpeg4:laser:2005	Occurs when a Timed Element is paused	Yes	No
“resumedEvent”	urn:mpeg:mpeg4:laser:2005	Occurs when a Timed Element is resumed	Yes	No
“activatedEvent”	urn:mpeg:mpeg4:laser:2005	Occurs when an element is activated, either as the result of an Activate command or a change in an active attribute to the value true, applied to it or to one of its ancestors.	No	No
“deactivatedEvent”	urn:mpeg:mpeg4:laser:2005	Occurs when an element is deactivated, either as the result of a Deactivate command or a change in an active attribute to the value false, applied to it or to one of its ancestors.	No	No
“screenOrientation0”	urn:mpeg:mpeg4:laser:2005	The screen orientation has changed to typical ‘landscape’ orientation (LAsEREvent)	No	No
“screenOrientation90”	urn:mpeg:mpeg4:laser:2005	The screen orientation has changed to typical ‘portrait’ orientation (LAsEREvent)	No	No
“screenOrientation180”	urn:mpeg:mpeg4:laser:2005	The screen orientation has changed to inverted ‘landscape’ orientation (LAsEREvent)	No	No
“screenOrientation270”	urn:mpeg:mpeg4:laser:2005	The screen orientation has changed to inverted ‘portrait’ orientation (LAsEREvent)	No	No
“stop”	urn:mpeg:mpeg4:laser:2005	Upon receiving such an event, a timed element behaves as if the uDOM method endElement() was called. (LAsEREvent)	No	No

Example:

Typical usage for screen orientation events is to have an animation triggered by one of these events, the animation changing the position/rotation of a group of scene elements to match the new screen orientation:

```
<animateTransform begin="lsr:screenOrientation90" to="..." xlink:href="#object1"
dur="1s"/>
<animateTransform begin="urn:mpeg:mpeg4:laser:2005:screenOrientation90" to="..."
xlink:href="#object2" dur="1s"/>
<animateTransform begin="urn:mpeg:mpeg4:laser:2005:screenOrientation90" to="..."
xlink:href="#object3" dur="1s"/>
...
<animateTransform begin="urn:mpeg:mpeg4:laser:2005:screenOrientation0" to="..."
xlink:href="#object1" dur="1s"/>
<animateTransform begin="urn:mpeg:mpeg4:laser:2005:screenOrientation0" to="..."
xlink:href="#object2" dur="1s"/>
<animateTransform begin="urn:mpeg:mpeg4:laser:2005:screenOrientation0" to="..."
xlink:href="#object3" dur="1s"/>
...
```

6.5.5. General IDL definition of LAsER events.

```
interface LAsEREvent : Event {
```

All LAsER specific events are prefixed with LAsER.

6.5.6. IDL Events definitions

6.5.6.1 ExternalValueEvent

```
interface ExternalValueEvent : LAsEREvent {
    readonly attribute float absoluteValue;
    readonly attribute boolean computableAsFraction;
    readonly attribute float fraction;
};
```

No defined constants

Attributes

- **absoluteValue:** This value represent the status of a resource of any kind, e.g. the remaining battery time.
- **computableAsFraction:** This value indicates whether a fraction can be computed from the absoluteValue.
- **fraction:** This value shall be between 0 and 1 inclusively and represent the status of the resource, e.g. the fraction of remaining battery time over operation time when fully charged.

No defined methods

Example: The following event could be defined:

"batteryState"	urn:example:X	Received by the document at system-dependent intervals and informs about the fraction of battery charge remaining. (ExternalValueEvent)	No	No
----------------	---------------	---	----	----

and used as follows:

```
<svg viewBox="0 0 1000 1000" baseProfile="tiny" id="root"
  xmlns="http://www.w3.org/2000/svg" version="1.1"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:X="urn:example:X">
  <rect width="100" height="100" stroke="blue"
    stroke-width="1">
    <animateColor id="myBatteryAnim" attributeName="fill"
      begin="indefinite" from="red" to="blue"
      dur="indefinite">
  </rect>
  <ev:listener observer="root" event="X:batteryState"
    handler="#myBatteryAnim">
</svg>
```

Add the following text to 6.7.1 before “The following restrictions apply to all commands”:

- **Activate:** to transfer an element from the waiting tree to the scene tree
- **Deactivate:** to transfer an element from the scene tree to the waiting tree
- **ReleaseResource:** to instruct the LAsER engine that a resource (typically media stream) will no longer be used in the scene and may therefore be reclaimed.

iTech STANDARD PREVIEW

Add the following text at the end of subclause 6.7.1:
(standards.iteh.ai)

A separate DOM tree is defined in addition to the scene tree: the waiting tree. The compositor and renderer have no knowledge of the waiting tree, thus objects in the waiting tree are neither composited nor rendered. The LAsER Commands are extended to search for elements first in the scene tree, then in the waiting tree if the elements are not found in the scene tree. The commands Activate and Deactivate are defined to operate on the waiting tree.

NOTE Elements placed in the waiting tree can be updated but are not accessible from the scene tree. A use element pointing to an element placed into the waiting tree behaves as if the referred element did not exist, i.e. nothing is rendered. Deactivating / activating a subtree with listeners may imply unregistering / registering the listeners observing elements not in the waiting tree.

Example:

```
<!-- Module 1 -->
<g id="module1">
  <rect x="10" y="10" width="176" height="20" fill="green"/>
  <text x="88" y="25" text-anchor="middle">Press FIRE</text>
  <ev:listener event="accessKey(FIRE)" handler="#go1"/>
  <ev:listener event="foo.click" handler="#go3"/>
  <animate xlink:href="#foo" .../>
  ...
</g>

<!-- Module 2 -->
<g id="module2">
  <rect x="0" y="10" width="176" height="20" fill="lime"/>
  <text x="88" y="25" text-anchor="middle">Press FIRE</text>
  <ev:listener event="accessKey(FIRE)" handler="#go2"/>
  <ev:listener event="foo.click" handler="#go4"/>
  <animate xlink:href="#foo" .../>
  ...
</g>
```

```

<g id="foo">
  ...
</g>

<!-- Modules aggregation and control -->
<lsr:conditional begin="0;accessKey(1)">
  <lsr:Deactivate ref="#module2"/>
  <lsr:Activate ref="#module1"/>
</lsr:conditional>
<lsr:conditional begin="accessKey(2)">
  <lsr:Deactivate ref="#module1"/>
  <lsr:Activate ref="#module2"/>
</lsr:conditional>

```

This simple example shows 2 independent modules. Since they use both the FIRE key, they cannot be both “active” at the same time. The two conditionals at the end implement very simply an exclusive activation of the two modules.

Add the following text at end of 6.7.10.1 (Save Command) Semantics:

Only attributes of elements identified by string IDs can be saved and restored reliably across scenes.

Attributes referencing elements by ID can only be saved and restored reliably across scenes if they hold a string ID.

In subclause 6.7.11.2, add a bullet:

- **intvalue2**: this attribute defines the second int value of the event.

In subclause 6.7.11.2, replace the table with:

<https://standards.iteh.ai/catalog/standards/sist/8dbba16d-bc1d-4309-9f33-4e254e/iso-iec-14496-20-2006-amd-1-2008>

Name of event	Attribute of event	Attribute of SendEvent
TimeEvent	detail	intvalue
UIEvent	detail	intvalue
WheelEvent	wheelDelta	intvalue
MouseEvent	screenX	<i>not carried</i>
	screenY	<i>not carried</i>
	clientX	pointvalue
	clientY	pointvalue
	button	intvalue
TextEvent	data	stringvalue
KeyEvent	keyIdentifier	when used with accessKey, longAccessKey, shortAccessKey and repeatKey, the key identifier follows the event name, e.g. accessKey(LEFT); with other key events, use stringvalue when unknown keyIdentifier, otherwise use intvalue with LAsER-defined keyIdentifier
ProgressEvent	lengthComputable	<i>not carried, the LAsER engine shall set this value to true if the value total is present</i>
	loaded	intvalue
	total	invalue2

After subclause 6.7.11, add the following subclauses:

6.7.12 Activate

6.7.12.1 Semantics

The Activate command reverses the effect of the Deactivate command, i.e. it restores the target element from the waiting tree to the scene tree. From the DOM point of view, the effect on the element is that the `Isr:ghost` element is replaced by the element, e.g. using a DOM `replaceChild`.

Note – The `Isr:ghost` element may have been transferred to the waiting tree as part of a Activate command applied to one of its ancestors.

6.7.12.2 Attributes

- **ref**: the id of the element which shall be restored from the waiting tree to the scene tree.

6.7.13 Deactivate

6.7.13.1 Semantics

The Deactivate command takes an element and places it into the waiting tree, so that it can be restored later to the scene tree by a Activate command. From the DOM point of view, the effect on the element is:

- the element is replaced by an `Isr:ghost` element, e.g. using a DOM `replaceChild`, regardless of whether the element is in the scene tree or in the waiting tree. Note: the element could already be in the waiting tree if one of its ancestors has been placed in the waiting tree.
- the element is placed in the waiting tree.

NOTE The `Isr:ghost` element is never encoded nor transmitted nor rendered.

[ISO/IEC 14496-20:2006/Amd 1:2008](https://standards.iteh.ai/catalog/standards/sist/8dbba16d-bc1d-4309-9f33-b4f0d9e2544e/iso-iec-14496-20-2006-amd-1-2008)

6.7.13.2 Attributes

- **ref**: the id of the element which shall be placed in the waiting tree.

6.7.14 ReleaseResource

6.7.14.1 Semantics

The ReleaseResource command instructs the LAsER engine that the indicated resource will no longer be used in the scene. All associated LAsER engine resources (memory, I/O) may be reclaimed. All scene elements that may be referencing this resource behave as if they pointed at a non available resource.

6.7.14.2 Attributes

- **ref**: this attribute defines the resource which may be discarded. It may be a stream ID or any other URI.

In subclause 6.8.1, replace sentence:

The list of possible attributes for an element is given in the summary table in subclause 6.8.53.

with:

The list of possible attributes for an element is given in the summary table in subclause 6.9.

Add to subclause 6.8.4.1:

When an animate element is not active and it receives an ExternalValueEvent, then the element sets the animation target attribute value as if the animation progress was at the fraction contained in the ExternalValueEvent.

When an animate element is not active and it receives a ProgressEvent which has lengthComputable=true, then the element sets the animation target attribute value as if the animation progress was at the fraction loaded over total.

Add to subclause 6.8.5.1:

When an animateColor element is not active and it receives an ExternalValueEvent, then the element sets the animation target attribute value as if the animation progress was at the fraction contained in the ExternalValueEvent.

When an animateColor element is not active and it receives a ProgressEvent which has lengthComputable=true, then the element sets the animation target attribute value as if the animation progress was at the fraction loaded over total.

Add to subclause 6.8.6.1:

When an animateMotion element is not active and it receives an ExternalValueEvent, then the element sets the animation target attribute value as if the animation progress was at the fraction contained in the ExternalValueEvent.

When an animateMotion element is not active and it receives a ProgressEvent which has lengthComputable=true, then the element sets the animation target attribute value as if the animation progress was at the fraction loaded over total.

<https://standards.iteh.ai/catalog/standards/sist/8dbba16d-bc1d-4309-9f33-b4f0d9e2544e/iso-iec-14496-20-2006-amd-1-2008>

Add to subclause 6.8.7.1:

When an animateTransform element is not active and it receives an ExternalValueEvent, then the element sets the animation target attribute value as if the animation progress was at the fraction contained in the ExternalValueEvent.

When an animateTransform element is not active and it receives a ProgressEvent which has lengthComputable=true, then the element sets the animation target attribute value as if the animation progress was at the fraction loaded over total.

Add to subclause 6.8.10.1:

The children of the conditional element are not accessible through DOM or LAsER Commands.

Add to subclause 6.8.28.2:

- **width:** the width of the clipping rectangle. This can also be accessed through the x component of the size field. This attribute is animatable but not inheritable.
- **height:** the height of the clipping rectangle. This can also be accessed through the y component of the size field. This attribute is animatable but not inheritable.
- **x:** the horizontal coordinate of the center of the clipping rectangle. This attribute is animatable but not inheritable.
- **y:** the vertical coordinate of the center of the clipping rectangle. This attribute is animatable but not inheritable.