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

**Part 20:
Lightweight Application Scene
Representation (LAsER) and Simple
Aggregation Format (SAF)**

**AMENDMENT 1: Extensions to support
SVGT1.2**

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 pour soutien de SVGT1.2

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.



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2009

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*.

Information technology — Coding of audio-visual objects —

Part 20:

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

AMENDMENT 1: Extensions to support SVGT1.2

In Clause 2, add the following reference:

W3C SVGT12, Scalable Vector Graphics (SVG) Tiny 1.2 Specification [Recommendation],
<http://www.w3.org/TR/SVGMobile12/>

Throughout the document, substitute references to [2] by references to [W3C SVGT12]

In Table 3, add these lines:

"screenOrientationLandscape"	urn:mpeg:mpeg4:laser:2005	The screen orientation has changed to typical "landscape" orientation Note – screenOrientationLandscape is sent either with screenOrientation0 or screenOrientation180	No	No
"screenOrientationPortrait"	urn:mpeg:mpeg4:laser:2005	The screen orientation has changed to typical "portrait" orientation Note – screenOrientationPortrait is sent either with screenOrientation90 or screenOrientation270	No	No

After 6.5.6.1, add the following subclause:

6.5.6.2 ScreenOrientationEvent

```
interface ScreenOrientationEvent : Event
{
    readonly attribute unsigned long screenWidth;
    readonly attribute unsigned long screenHeight;
    readonly attribute unsigned long screenAngle;
}
```

No defined constants.

Attributes:

screenWidth - contains the new screen display or viewport width

screenHeight - contains the new screen display or viewport height

screenAngle - documents the angle from the horizontal positive x axis to the axis of the longest dimension of the screen, measured clockwise.

No defined methods.

The orientation is reported in degrees in `screenAngle`, to the best of the terminal's capability. This is measured as the angle between the positive X-axis of an un-rotated frame of reference, and the orientation of the longer of the X or Y axis of the screen, as rotated, as shown in Figure AMD2.1. Specifically, for a screen that is normally portrait and in its normal position, the `screenAngle` is 90 degrees, since the longest axis is vertical, and the Y-axis is downward in SVG. Similarly, for a terminal that is normally landscape and in its normal position, the `screenAngle` is 0 degrees, since the longest axis is horizontal, and is the X-axis. This angle therefore would normally be close to 90 or 270 in portrait events, and close to 0 or 180 in landscape events, and 0 or 90 in terminals that are in their normal orientation, and 180 or 270 in terminals that are inverted.

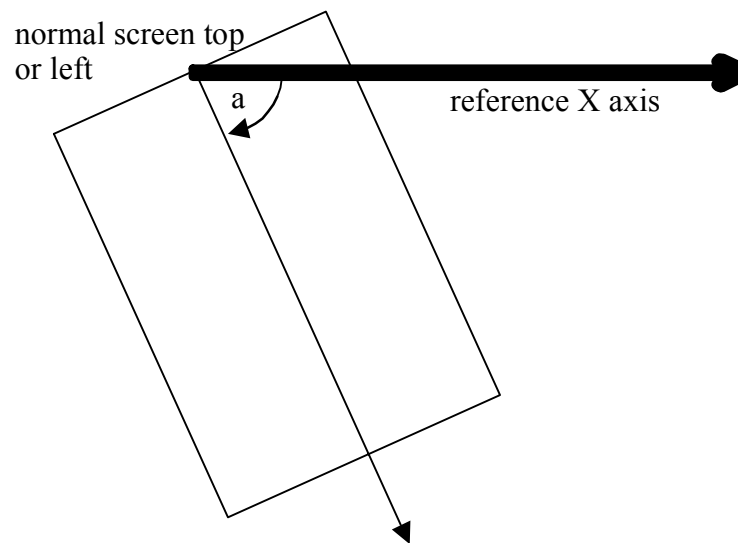


Figure AMD2.1 — Angle measurement procedure

In 6.6.2.2, add at the end:

Note – Data is saved as a set of four values, using the URI, `pathComponents` and `useFullRequestHost` from the stream containing the save command:

- the domain-name formed from the URI and `useFullRequestHost`;
- the path formed from the URI and `pathComponents`;
- the groupID (known as 'name' in [RFC 2965](#));
- the set of {element-ID, attribute-name, value} triplets.

When a restore command is executed all saved sets with the same (equal) groupID, and also where the URI of the stream containing the restore command matches the saved domain-name and path, are restored. This matching is defined in section 4.3.4 of [RFC 2965](#).

The same matching rules are used for the clean command.

In 6.7.1, add at the end of the paragraph on the waiting tree:

Scripts also have access to the waiting tree in the same way as LAsER Commands.

In 6.7.3.1, add at the end:

The Clean command behaves exactly the same as a Save command that saves no state; as is normal for the Save command, any other saved state with the same domain-name, path, and groupId is replaced, in this case, with an empty set of saved data. This is functionally equivalent to deleting that saved state, as nothing would be restored.

In 6.7.6.4, replace `document.svgDocument.root` with `document.documentElement`

In 6.7.7.4, replace `document.svgDocument.root` with `document.documentElement`

In 6.7.10.1, add at the end:

The Save command saves the values of a set of attributes, each identified by element ID and attribute name. Each save operation uses a groupId. Any other saved state with the same domain-name, path, and groupId is replaced.

In 6.7.11.2, add to the event bullet:

The XML syntax for this attribute with events `accessKey`, `longAccessKey`, `shortAccessKey` and `repeatKey` includes the reference key name between parentheses, e.g. **`event="accessKey(FIRE)"`**, regardless of whether the key code is defined in LAsERML or not.

After 6.7.14, add the following new subclauses:

6.7.15 Immediate Script Execution

6.7.15.1 Semantics

The `doScript` command supplies a script for immediate execution as specified in [3GPP DIMS].

The following example retrieves the `rect` element with id `"myRect"` child of the root `svg` element, removes it from the document tree and append it in another position as a child of a `g` element with id `"myGroup"`:

```
<doScript type="application/ecmascript">
  var myrect = document.getElementById("myRect");
  document.documentElement.removeChild(myrect);
  document.getElementById("myGroup").appendChild(myrect);
</doScript>
```

6.7.15.2 Attributes

`type` - is a string that identifies the scripting language used. It takes a suitable MIME type [19] from the IANA registry, such as `"application/ecmascript"` (see [14]).