INTERNATIONAL STANDARD

ISO/IEC 23005-7

Fourth edition
2019-12

Information technology — Media context and control —

Part 7: Conformance and reference software

Technologies de l'information — Contrôle et contexte de supports —

Partie 7: Conformité et logiciel de référence

Reference number
ISO/IEC 23005-7:2019(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see http://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This fourth edition cancels and replaces the third edition (ISO/IEC 23005-7:2017), which has been technically revised. The main changes compared to the previous edition are as follows:

— modification of introduction;

— modification of schema namespace.

A list of all parts in the ISO/IEC 23005 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

The ISO/IEC 23005 series provides an architecture and specifies information representation of data flowing in and out of the real world and virtual worlds.

The data for the real world are communicated through sensors and actuators. The data for virtual worlds consist of properties of virtual objects and multi-sensorial data embedded in audio-visual content. The ISO/IEC 23005 series specifies data formats for sensors, actuators, virtual objects, and audio-visual content.

Data captured from the real world could need to be adapted for use in a virtual world and data from virtual worlds could also need to be adapted for use in the real world. The ISO/IEC 23005 series does not specify how the adaptation is carried out but only specifies the interfaces:

— data for sensors are sensor capabilities, sensed data, and sensor adaptation preferences;

— data for actuators are sensory device capabilities, sensory device commands, and sensory effect preferences;

— data for virtual objects are characteristics of avatars and virtual world objects;

— sensory effect could be needed to enrich audio-visual contents.

The conformance and reference software is available at https://standards.iso.org/iso-iec/23005/-7/ed-4/en.

Information technology — Media context and control —

Part 7:
Conformance and reference software

1 Scope

This document specifies the conformance and reference software implementing the normative clauses of all parts of the ISO/IEC 23005 series. The information provided is applicable for determining the reference software modules available for the parts of the ISO/IEC 23005 series, understanding the functionality of the available reference software modules, and utilizing the available reference software modules. The available reference software modules are specified in the form of application programming interfaces (API) according to ISO/IEC 23006-1.

Furthermore, this document provides the means for conformance testing, i.e. bitstreams – XML descriptions – that conform or do not conform to the normative clauses of the other parts of the ISO/IEC 23005 series and informative descriptions thereof.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 23005-2, *Information technology — Media context and control — Part 2: Control information*

ISO/IEC 23005-3, *Information technology — Media context and control — Part 3: Sensory information*

ISO/IEC 23005-4, *Information technology — Media context and control — Part 4: Virtual world object characteristics*

ISO/IEC 23005-5, *Information technology — Media context and control — Part 5: Data formats for interaction devices*

ISO/IEC 23005-6, *Information technology — Media context and control — Part 6: Common types and tools*

ISO/IEC 23006-1, *Information technology — Multimedia service platform technologies — Part 1: Architecture*

W3C, Extensible Markup Language (XML) 1.0 (Fifth Edition) W3C Recommendation 26 November 2008. Available at https://www.w3.org/TR/xml/

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 23006-1 apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.2 Abbreviated terms

API application programming interface

CIM control information metadata

MXM MPEG extensible middleware

SEM sensory effect metadata

VWOC virtual world object characteristics

IIM interaction information metadata

4 Reference software for the ISO/IEC 23005 series

4.1 General

This clause specifies the reference software for the ISO/IEC 23005 series. The ISO/IEC 23005 series reference software is written in Java[1], which is a general purpose programming language and adopts the following package structure:

Name	*Definition*
`org`	Java package name for reference software provided by organizations such as ISO/IEC, W3C, or similar.
`org.iso`	Java package name for reference software provided by ISO/IEC.
`org.iso.mpeg`	Java package name for reference software provided by ISO/IEC.
`org.iso.mpeg.mpegv`	Java package name for reference software provided in the course of the development of the ISO/IEC 23005 series. NOTE 1 Subsequent packages for the individual parts of the ISO/IEC 23005 series use the uncapitalized abbreviations as defined in 3.2, e.g., — `org.iso.mpeg.mpegv.cidl`, `org.iso.mpeg.mpegv.dcdv`, `org.iso.mpeg.mpegv.sapv`, `org.iso.mpeg.mpegv.scdv` and `org.iso.mpeg.mpegv.sepv` for ISO/IEC 23005-2, — `org.iso.mpeg.mpegv.sedl` and `org.iso.mpeg.mpegv.sev` for ISO/IEC 23005-3, — `org.iso.mpeg.mpegv.vwoc` for ISO/IEC 23005-4, or — `org.iso.mpeg.mpegv.iidl`, `org.iso.mpeg.mpegv.dcv` and `org.iso.mpeg.mpegv.siv` for ISO/IEC 23005-5. — `org.iso.mpeg.mpegv.ct` for ISO/IEC 23005-6. NOTE 2 Code included within these packages can be generated automatically by using tools such as JAXB (Java Architecture for XML Binding).
`org.iso.mpeg.mpegv.{part}.binary`	Java package name for binary representation of reference software provided in the course of the development of the ISO/IEC 23005 series. the {`part`} placeholder is replaced by the component of each part, i.e., `cidl`, `dvdv`, `sapv`, `scdv`, `sepv`, `sedl`, `sev`, `vwoc`, `iidl`, `dcv`, `siv` and `ct`.
`org.iso.mpeg.mxm`	Java package name for reference software provided in the course of the development of ISO/IEC 23006-1. The actual API is defined within this package.

4.2 ISO/IEC 23005-2 APIs

4.2.1 General

This subclause specifies the API to the ISO/IEC 23005-2 reference software. The API is defined in Java and adopts the following package structure.

Name	*Definition*
`org.iso.mpeg.mxm.engine.cimengine`	Java package name for API to the ISO/IEC 23005-2 reference software.

4.2.2 CIM engine

The CIMEngine interface defines the methods for operating over data structures as defined within ISO/IEC 23005-2. Classes implementing the CIM engine interface act as factories creating instances of classes performing the following functionalities:

— classes to create data structures, by means of the CIM creation engine;

— classes to access data contained in data structures, by means of CIM parser engine.

4.2.3 CIM creation

Creating CIM structures involves the following interfaces:

— CIMCreator: an interface defining the methods to create CIM structures.

4.2.4 CIM access

Accessing CIM structures involves the following interfaces:

— CIMParser: an interface defining the methods to parse CIM structures.

4.3 ISO/IEC 23005-3 APIs

4.3.1 General

This subclause specifies the API to the ISO/IEC 23005-3 reference software. The API is defined in Java and adopts the following package structure.

Name	*Definition*
`org.iso.mpeg.mxm.engine.semengine`	Java package name for API to the ISO/IEC 23005-3 reference software.

4.3.2 SEM engine

The SEMEngine interface defines the methods for operating over data structures as defined within ISO/IEC 23005-3. Classes implementing the SEM engine interface act as factories creating instances of classes performing the following functionalities:

— classes to create data structures, by means of the SEM creation engine;

— classes to access data contained in data structures, by means of SEM parser engine.

4.3.3 SEM creation

Creating SEM structures involves the following interfaces:

— SEMCreator: an interface defining the methods to create SEM structures.

4.3.4 SEM access

Accessing SEM structures involves the following interfaces:

— SEMParser: an interface defining the methods to parse SEM structures.

4.4 ISO/IEC 23005-4 APIs

4.4.1 General

This subclause specifies the API to the ISO/IEC 23005-4 reference software. The API is defined in Java and adopts the following package structure.

Name	*Definition*
`org.iso.mpeg.mxm.engine.vwocengine`	Java package name for API to the ISO/IEC 23005-4 reference software.

4.4.2 VWOC engine

The VWOCEngine interface defines the methods for operating over data structures as defined within ISO/IEC 23005-4. Classes implementing the VWOC engine interface act as factories creating instances of classes performing the following functionalities:

— classes to create data structures, by means of the VWOC creation engine;

— classes to access data contained in data structures, by means of VWOC parser engine.

4.4.3 VWOC creation

Creating VWOC structures involves the following interfaces:

— VWOCCreator: an interface defining the methods to create VWOC structures.

4.4.4 VWOC access

Accessing VWOC structures involves the following interfaces:

— VWOCParser: an interface defining the methods to parse VWOC structures.

4.5 ISO/IEC 23005-5 APIs

4.5.1 General

This subclause specifies the API to the ISO/IEC 23005-5 reference software. The API is defined in Java and adopts the following package structure.

Name	*Definition*
`org.iso.mpeg.mxm.engine.iidengine`	Java package name for API to the ISO/IEC 23005-5 reference software.

4.5.2 IID engine

The IIDEngine interface defines the methods for operating over data structures as defined within ISO/IEC 23005-2. Classes implementing the IID engine interface act as factories creating instances of classes performing the following functionalities:

— classes to create data structures, by means of the IID creation engine;

— classes to access data contained in data structures, by means of IID parser engine.

4.5.3 IID creation

Creating IID structures involves the following interfaces:

— IIDCreator: an interface defining the methods to create IID structures.

4.5.4 IID access

Accessing IID structures involves the following interfaces:

— IIDParser: an interface defining the methods to parse IID structures.

4.6 Binary representation APIs for the ISO/IEC 23005 series

4.6.1 General

This subclause specifies the API to the ISO/IEC 23005 series reference software for binary representation. The API is defined in Java and adopts the following package structure.

Name	*Definition*
`org.iso.mpeg.mpegv.binary.core`	Java package name for API to the ISO/IEC 23005 series reference software for binary representation.

4.6.2 BinaryIO

The BinaryIO interface defines the methods for operating over data structures as defined within the ISO/IEC 23005 series. Classes implementing the BinaryIO interface act as factories creating instances of classes performing the following functionalities:

— classes to generate a binary structure from a MXM object tree, by means of the construct binary structure function;

— classes to generate a binary file from a binary structure, by means of the write binary function;

— classes to generate a binary structure from a binary structure, by means of the read binary function;

— classes to restore a MXM object tree from a binary structure, by means of the create MXM object function.

4.6.3 DefaultBinaryIO

DefaultBinaryIO class defines the functions “encode” and “decode”. Root elements of other parts (e.g. ISO/IEC 23005-2, ISO/IEC 23005-3, ISO/IEC 23005-4) can be extended from this DefaultBinaryIO class. DefaultBinaryIO involves the following functions:

— encode: a function defining the methods to create binary file from a MXM object tree;

— decode: a function defining the methods to restore a MXM object tree from a binary file.

5 Conformance for the ISO/IEC 23005 series

5.1 General

This clause defines conformance for the ISO/IEC 23005 series. A bitstream is compliant to the ISO/IEC 23005 series if it conforms to the conformance definitions specified in 5.2 and 5.3.

5.2 Rule-based conformance for the ISO/IEC 23005 series

5.2.1 General

This subclause defines a validation schema based on ISO/IEC 19757-3 to check the conformance bitstreams – XML descriptions – that conform or do not conform to the normative clauses of the other parts of the ISO/IEC 23005 series and descriptions thereof. In particular, the additional validation rules as specified in the other parts of the ISO/IEC 23005 series are checked with this validation schema. The actual schema can be found in 5.2.2 and a more readable version of the rules and assertion messages is provided in 5.2.3. The actual conformance bitstreams – XML descriptions – are defined in 5.2.4. The actual conformance bitstreams and corresponding code are available at https://standards.iso.org/iso-iec/23005/-7/ed-4/en.

For transforming the validation schema according to ISO/IEC 19757-3[2] and for validating the examples, Saxon[1] is used. Saxon is not mandatory and other tools can be used for validating the examples.

5.2.2 Validation schema


```
<schema
  xmlns="http://purl.oclc.org/dsdl/schematron"
  xmlns:sedl="urn:mpeg:mpeg-v:2018:01-SEDL-NS"
  xmlns:sev="urn:mpeg:mpeg-v:2018:01-SEV-NS"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:si="urn:mpeg:mpeg21:2003:01-DIA-XSI-NS"
  xmlns:cidl="urn:mpeg:mpeg-v:2018:01-CIDL-NS"
  xmlns:dcdv="urn:mpeg:mpeg-v:2018:01-DCDV-NS"
  xmlns:scdv="urn:mpeg:mpeg-v:2018:01-SCDV-NS"
  xmlns:sepv="urn:mpeg:mpeg-v:2018:01-SEPV-NS"
  xmlns:mpegvct="urn:mpeg:mpeg-v:2018:01-CT-NS"
  xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004"
  xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS"
  xmlns:iidl="urn:mpeg:mpeg-v:2018:01-IIDL-NS"
  xmlns:dcv="urn:mpeg:mpeg-v:2018:01-DCV-NS"
  xmlns:siv="urn:mpeg:mpeg-v:2018:01-SIV-NS"
  queryBinding='xslt' schemaVersion='ISO19757-3'>
  <ns prefix="sedl" uri="urn:mpeg:mpeg-v:2018:01-SEDL-NS"/>
  <ns prefix="si" uri="urn:mpeg:mpeg21:2003:01-DIA-XSI-NS"/>
  <ns prefix="sev" uri="urn:mpeg:mpeg-v:2018:01-SEV-NS"/>
  <ns prefix="xsi" uri="http://www.w3.org/2001/XMLSchema-instance"/>
  <ns prefix="mpeg7" uri="urn:mpeg:mpeg7:schema:2004"/>
  <ns prefix="cidl" uri="urn:mpeg:mpeg-v:2018:01-CIDL-NS"/>
  <ns prefix="dcdv" uri="urn:mpeg:mpeg-v:2018:01-DCDV-NS"/>
  <ns prefix="scdv" uri="urn:mpeg:mpeg-v:2018:01-SCDV-NS"/>
  <ns prefix="sepv" uri="urn:mpeg:mpeg-v:2018:01-SEPV-NS"/>
  <ns prefix="mpegvct" uri="urn:mpeg:mpeg-v:2018:01-CT-NS"/>
  <ns prefix="dia" uri="urn:mpeg:mpeg21:2003:01-DIA-NS"/>
  <ns prefix="iidl" uri="urn:mpeg:mpeg-v:2018:01-IIDL-NS"/>
  <ns prefix="dcv" uri="urn:mpeg:mpeg-v:2018:01-DCV-NS"/>
  <ns prefix="siv" uri="urn:mpeg:mpeg-v:2018:01-SIV-NS"/>
  <title>Schema for validating SEDL/CIDL/IIDL descriptions</title>
  <!-- ##################################### -->
  <!-- Schema for validating CIDL(Part 2) descriptions -->
  <!-- ##################################### -->
  <pattern name="SensorDeviceCapability element">
    <!-- R1.0: Check the conformance of SensorDeviceCapability -->
    <rule context="cidl:SensorDeviceCapability">
      <assert test="if (@minValue and @maxValue) then if (@minValue > @maxValue) then
false() else true() else true()">
```



```
        A minValue shall be less than or equal to a maxValue.
      </assert>
    </rule>
  </pattern>
  <pattern name="PositionCapability element">
    <!-- R1.1: Check the conformance of PositionCapability -->
    <rule context="scdv:PositionCapability">
      <assert test="if (@minValue and @maxValue) then if (@minValue > @maxValue) then
false() else true() else true()">
        A minValue shall be less than or equal to a maxValue.
      </assert>
    </rule>
  </pattern>
  <pattern name="OrientationCapability element">
    <!-- R1.2: Check the conformance of OrientationCapability -->
    <rule context="scdv:OrientationCapability">
      <assert test="if (@minValue and @maxValue) then if (@minValue > @maxValue) then
false() else true() else true()">
        A minValue shall be less than or equal to a maxValue.
      </assert>
    </rule>
  </pattern>
  <pattern name="VelocityCapability element">
    <!-- R1.3: Check the conformance of VelocityCapability -->
    <rule context="scdv:VelocityCapability">
      <assert test="if (@minValue and @maxValue) then if (@minValue > @maxValue) then
false() else true() else true()">
        A minValue shall be less than or equal to a maxValue.
      </assert>
    </rule>
  </pattern>
  <pattern name="AngularVelocityCapability element">
    <!-- R1.4: Check the conformance of AngularVelocityCapability -->
    <rule context="scdv:AngularVelocityCapability">
      <assert test="if (@minValue and @maxValue) then if (@minValue > @maxValue) then
false() else true() else true()">
        A minValue shall be less than or equal to a maxValue.
      </assert>
    </rule>
  </pattern>
  <pattern name="AccelerationCapability element">
    <!-- R1.5: Check the conformance of AccelerationCapability -->
    <rule context="scdv:AccelerationCapability">
      <assert test="if (@minValue and @maxValue) then if (@minValue > @maxValue) then
false() else true() else true()">
        A minValue shall be less than or equal to a maxValue.
      </assert>
    </rule>
  </pattern>
  <pattern name="AngularAccelerationCapability element">
    <!-- R1.6: Check the conformance of AngularAccelerationCapability -->
    <rule context="scdv:AngularAccelerationCapability">
      <assert test="if (@minValue and @maxValue) then if (@minValue > @maxValue) then
false() else true() else true()">
        A minValue shall be less than or equal to a maxValue.
      </assert>
    </rule>
  </pattern>
  <pattern name="Range element">
    <!-- R2.*: Check the conformance of range -->
    <rule context="scdv:Range">
      <!-- R2.0 -->
      <assert test="if (child::scdv:XminValue > child::scdv:XmaxValue) then false() else
true()">
        An XminValue shall be less than or equal to an XmaxValue.
      </assert>
      <!-- R2.1 -->
      <assert test="if (child::scdv:YminValue > child::scdv:YmaxValue) then false() else
true()">
        An YminValue shall be less than or equal to an YmaxValue.
      </assert>
```