
**Information technology — Multimedia
framework (MPEG-21) —**

**Part 14:
Conformance Testing**

*Technologies de l'information — Cadre multimédia (MPEG-21) —
Partie 14: Essais de conformité*



Reference number
ISO/IEC 21000-14:2007(E)

AVA Documents
Document Sharing Hub
© ISO/IEC 2007

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 2007

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

Contents

Page

Foreword.....	vii
1 Scope	1
2 Normative references	1
3 DID Conformance.....	2
3.1 DIDL Document Conformance.....	2
3.2 DIDL Parser Conformance	2
4 DII Conformance	2
4.1 Digital Item Identification Document Conformance	2
4.2 Digital Item Identification Parser Conformance	2
5 IPMP Components Conformance	2
5.1 Digital Item Declaration Document Conformance	2
5.2 IPMP Parser Conformance.....	3
6 IPMP Amd.1 Conformance	3
6.1 IPMP DIDL	3
6.2 IPMP Conformance Parser.....	3
7 REL Conformance.....	3
7.1 Rights Expression Conformance	3
7.1.1 General.....	3
7.1.2 Rules extracted from subclause 6.3 of ISO/IEC 21000-5:2004	3
7.1.3 Rules extracted from subclauses 6.5.2 and 6.6 of ISO/IEC 21000-5:2004.....	5
7.1.4 Rules extracted from subclause 7.4.1 of ISO/IEC 21000-5:2004	6
7.1.5 Rules extracted from subclause 7.3.6 of ISO/IEC 21000-5:2004	8
7.2 REL Parser Conformance	8
8 REL Amd.1 Conformance	8
8.1 Introduction	8
8.2 Rules extracted from subclause 6.3 of ISO/IEC 21000-5:2004	8
8.3 Rules extracted from subclauses 6.5.2 and 6.6 of ISO/IEC 21000-5:2004.....	9
8.4 Rules extracted from subclause 7.4.1 of ISO/IEC 21000-5:2004	10
8.5 Rules for the DrmSystem condition.....	10
9 REL Amd.2 Conformance	10
9.1 Introduction	10
9.2 LicensePartId and LicensePartIdRef	10
9.3 Variable reference and conceptually abstract elements and types	14
9.4 LicensePart optional content model.....	16
10 RDD Conformance	17
11 DIA Conformance.....	17
11.1 DIA Description Conformance.....	17
11.2 Bitstream Syntax Schema Conformance	17
11.3 DIA Parser Conformance	17
11.4 DIA BintobSD Parser Conformance	18
11.5 DIA BSDtoBin Parser Conformance	18
11.6 DIA gBSDtoBin Parser Conformance	19
11.7 DIA gBSDtoAU Conformance	19
12 DIA Amd.1 Conformance	20
13 DIA Amd.2 Conformance	20
13.1 Simple_Mode_XML_Fragmenter	20

14	File Format Conformance.....	20
14.1	Introduction	20
14.2	Process	20
14.3	Areas tested.....	20
14.4	File Documentation.....	20
15	DIP Conformance	21
15.1	Introduction	21
15.2	DIP schema conformance	21
15.2.1	Introduction	21
15.2.2	Testing DIP schema conformance	21
15.2.3	DIP engine conformance	22
15.2.4	Introduction	22
15.2.5	Testing DIP engine conformance	22
15.3	DIML syntax conformance	23
15.3.1	Introduction	23
15.3.2	Testing DIML syntax conformance.....	23
15.4	DIM engine conformance	24
15.4.1	Introduction	24
15.4.2	DIBOs with non-deterministic results	24
15.4.3	Testing DIM engine conformance.....	26
15.5	J-DIXO conformance.....	27
15.5.1	Introduction	27
15.5.2	J-DIXO inclusion in DID Conformance.....	27
15.5.3	J-DIXO conformance.....	27
16	DIP Amd.1 Conformance	28
16.1	Terminal conformance: C++ binding implementation.....	28
16.2	C++ executable conformance	28
17	Event Reporting Conformance	28
17.1	Event Report Request Description Conformance	28
17.2	Event Report Description Conformance.....	28
17.3	Event Report Request Parser Conformance	28
17.4	Event Report Parser Conformance	29
17.5	Draft List of Conformance Tests	29
18	Binary Format Conformance.....	29
19	Fragment Identification Conformance	30
19.1	Introduction	30
19.2	Syntax Conformance	30
19.3	FID Parser conformance.....	30
19.4	Logical Model conformance.....	30
19.5	Logical Model Parser conformance	30
Annex A (normative) Tests for DIDL Documents	31	
A.1	Introduction	31
A.2	Test 1	31
A.3	Test 2	33
A.4	Test 3	34
A.5	Test 4	34
A.6	Test 5	36
A.7	Test 6	36
A.8	Test 7	37
A.9	Test 8	37
A.10	Test 9	38
A.11	Test 10	39
A.12	Test 11	40
A.13	Test 12	40
A.14	Test 13	41
A.15	Test 14	42
A.16	Test 15	42

A.17	Test 16.....	43
A.18	Test 17.....	44
A.19	Test 18.....	44
A.20	Test 19.....	45
A.21	Test 20.....	45
A.22	Test 21.....	46
A.23	Test 22.....	46
A.24	Test 23.....	47
A.25	Test 24.....	48
A.26	Test 25.....	48
A.27	Test 26.....	49
A.28	Test 27.....	50
A.29	Test 28.....	50
Annex B	(normative) Tests for Digital Item Identification Documents	52
B.1	Test 1.....	52
B.2	Test 2.....	52
B.3	Test 3.....	52
B.4	Test 4.....	53
Annex C	(normative) Tests for DIDL Documents with protected elements	54
C.1	Introduction	54
C.2	Test 1.....	54
C.3	Test 2.....	56
C.4	Test 3.....	57
C.5	Test 4.....	59
C.6	Test 5.....	60
Annex D	(normative) Tests for IPMP Amd.1	62
D.1	Introduction	62
D.2	Test 1.....	62
D.3	Test 2.....	63
D.4	Test 3.....	64
D.5	Test 4.....	65
D.6	Test 5.....	66
D.7	Test 6.....	67
Annex E	(normative) Tests for Rights Expressions	69
E.1	Test 1.....	69
E.2	Test 2.....	70
E.3	Test 3.....	71
E.4	Test 4.....	71
E.5	Test 5.....	72
E.6	Test 6.....	73
E.7	Test 7.....	74
E.8	Test 8.....	74
E.9	Test 9.....	75
E.10	Test 10.....	76
E.11	Test 11.....	76
E.12	Test 12.....	77
E.13	Test 13.....	77
E.14	Test 14.....	78
E.15	Test 15.....	79
E.16	Test 16.....	79
E.17	Test 17.....	80
E.18	Test 18.....	80
E.19	Test 19.....	81
E.20	Test 20.....	81
E.21	Test 21.....	82
E.22	Test 22.....	82
E.23	Test 23.....	83
E.24	Test 24.....	84

E.25	Test 25	84
E.26	Test 26	85
E.27	Test 27	85
E.28	Test 28	86
E.29	Test 29	87
Annex F	(normative) Tests for REL Amd.1	88
F.1	Test 1	88
F.2	Test 2	89
F.3	Test 3	89
F.4	Test 4	90
F.5	Test 5	91
F.6	Test 6	91
F.7	Test 7	92
F.8	Test 8	93
F.9	Test 9	94
F.10	Test 10	94
F.11	Test 11	95
F.12	Test 12	96
F.13	Test 13	96
F.14	Test 14	97
Annex G	(normative) Tests for REL Amd.2	98
Annex H	(normative) Tests for DIA Descriptions	99
Annex I	(normative) Tests for DIA Amd.1	112
I.1	Introduction	112
I.2	Conformance for Conversion Link	112
I.3	Conformance for Composite Conversion	114
I.4	Conformance for Cross Conversion QoS	116
I.5	Conformance for Conversion URI	118
Annex J	(normative) Tests for DIA Amd.2	121
J.1	Simple_Mode_XML_Fragmenter	121
Annex K	(normative) Tests for File Format	122
K.1	Areas tested	122
Annex L	(normative) Tests for DIP	123
L.1	Tests for DIP engine	123
L.2	Tests for DIM engine	123
Annex M	(normative) Tests for DIP Amd.1	124
M.1	C++ data type bindings for DIML object types	124
M.2	C++ DIBO factory interface	124
M.3	C++ global environment interface	124
M.4	C++ interface bindings for DIBOs	124
Annex N	(normative) Tests for ER	125
N.1	Introduction	125
Annex O	(normative) Tests for Binary Format	126
O.1	Introduction	126
Annex P	(normative) Tests for FID	127
P.1	Test Fragment Identifiers	127
P.2	Test Logical Model Schemas	130

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.

ISO/IEC 21000-14 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

ISO/IEC 21000 consists of the following parts, under the general title *Information technology — Multimedia framework (MPEG-21)*:

- *Part 1: Vision, Technologies and Strategy* [Technical Report]
- *Part 2: Digital Item Declaration*
- *Part 3: Digital Item Identification*
- *Part 4: Intellectual Property Management and Protection Components*
- *Part 5: Rights Expression Language*
- *Part 6: Rights Data Dictionary*
- *Part 7: Digital Item Adaptation*
- *Part 8: Reference Software*
- *Part 9: File Format*
- *Part 10: Digital Item Processing*
- *Part 11: Evaluation Tools for Persistent Association Technologies* [Technical Report]
- *Part 12: Test Bed for MPEG-21 Resource Delivery* [Technical Report]
- *Part 14: Conformance Testing*
- *Part 15: Event Reporting*
- *Part 16: Binary Format*
- *Part 17: Fragment Identification of MPEG Resources*
- *Part 18: Digital Item Streaming*

Information technology — Multimedia framework (MPEG-21) —

Part 14: Conformance Testing

1 Scope

This International Standard specifies conformance points and conformance tests for different parts of ISO/IEC 21000. Based on the various conformance points, it is identified which requirements defined in ISO/IEC 21000 apply to those conformance points. The tests are developed to ascertain whether a particular artifact (such as a piece of software or hardware or a document) meets all the requirements for a specific conformance point or not.

ISO/IEC 21000-14:2007 specifies conformance points and tests for ISO/IEC 21000-2:2005, ISO/IEC 21000-3:2003, ISO/IEC 21000-4:2006, ISO/IEC 21000-4:2006/Amd.1:2007, ISO/IEC 21000-5:2004, ISO/IEC 21000-5:2004/Amd.1:2007, ISO/IEC 21000-5:2004/Amd.2:2007, ISO/IEC 21000-7:2004, ISO/IEC 21000-7:2004/Amd.1:2006, ISO/IEC 21000-7:2004/Amd.2:2007, ISO/IEC 21000-9:2005, ISO/IEC 21000-10:2006, ISO/IEC 21000-10:2006/Amd.1:2006, ISO/IEC 21000-15:2006, ISO/IEC 21000-16:2005, and ISO/IEC 21000-17:2006.

2 Normative references

The following referenced documents are indispensable for the application 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 21000 (all parts), *Information technology – Multimedia framework (MPEG-21)*

Namespaces in XML, W3C Recommendation, 14 January 1999

Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, 6 October, 2000

XML Schema Part 1: Structures, W3C Recommendation, May 2, 2001

XML Schema Part 2: Datatypes, W3C Recommendation, May 2, 2001

XML Information Set, W3C Recommendation, 24 October 2001

XML Base, W3C Recommendation, 27 June 2001

Canonical XML, Version 1.0, W3C Recommendation, 15 March 2001

3 DID Conformance

3.1 DIDL Document Conformance

A DIDL Document is conformant only if

- 1) It is a well-formed XML document that validates against the Digital Item Declaration Language Schema as specified in ISO/IEC 21000-2:2005, and
- 2) All validation rules that are normatively defined in ISO/IEC 21000-2:2005 are met.

NOTE 1 Conformance of DIDL Documents can be checked by using the normative Digital Item Declaration parser that is part of the ISO/IEC 21000 Reference Software.

NOTE 2 A DID may contain descriptions defined in other parts of ISO/IEC 21000, such as a DIA description (ISO/IEC 21000-7:2004). The conformance of a DID does not imply the conformance of these included descriptions.

3.2 DIDL Parser Conformance

A DIDL Parser is defined as a module that takes a file as input and produces as output either 1 (if the input file is a conformant DIDL Document) or 0 (if the input file is not a conformant DIDL Document).

A DIDL Parser is conformant only if, given the inputs given in Annex A it produces the corresponding outputs from Annex A.

4 DIJ Conformance

4.1 Digital Item Identification Document Conformance

A Digital Item Identification Document is conformant only if it is a well-formed XML document that validates against the Digital Item Identification Schema as specified in ISO/IEC 21000-3:2003.

4.2 Digital Item Identification Parser Conformance

A Digital Item Identification Parser is defined as a module that takes a file as input and produces as output either 1 (if the input file is a conformant Digital Item Identification Document) or 0 (if the input file is not a conformant Digital Item Identification Document).

A Digital Item Identification Parser is conformant only if, given the inputs given in Annex B it produces the corresponding outputs from Annex B.

5 IPMP Components Conformance

5.1 Digital Item Declaration Document Conformance

A DIDL document with IPMP DIDL elements is conformant only if

- 1) It is a well-formed XML document that validates against the DIDL Schema, IPMP DIDL Schema, IPMP Information Schema and IPMP General Information Schema as specified in ISO/IEC 21000-4:2006, and
- 2) All validation rules that are normatively defined in ISO/IEC 21000-4:2006 are met, and

- 3) Once unprotected the DIDL document shall be DID Conformant as specified in clause 3.

5.2 IPMP Parser Conformance

An IPMP Parser is defined as a module that takes a file as input and produces as output either 1 (if the input file is a conformant DIDL document with IPMP DIDL elements) or 0 (if the input file is not a conformant DIDL document with IPMP DIDL elements).

An IPMP Parser is conformant only if, given the inputs given in Annex C, it produces the corresponding outputs from Annex C.

6 IPMP Amd.1 Conformance

6.1 IPMP DIDL

The conformance of IPMP DIDL of ISO/IEC 21000-4:2006/Amd.1:2007 is as defined in ISO/IEC 21000-14 subclause 5.1 and additional rule:

All restrictions that are normatively defined in ISO/IEC 21000-4:2006/Amd.1:2007 subclauses 9.3.3, 9.3.4, and 9.3.5 are met.

6.2 IPMP Conformance Parser

An IPMP Base Profile Parser is defined as a module that takes a file as input and produces as output either 1 (if the input file is a conformant DIDL document with IPMP DIDL elements) or 0 (if the input file is not a conformant DIDL document with IPMP DIDL elements).

An IPMP Parser is conformant only if, given the inputs given in Annex D, it produces the corresponding outputs from Annex D.

7 REL Conformance

7.1 Rights Expression Conformance

7.1.1 General

A Rights Expression is conformant only if it meets all the Rights Expression constraints specified in ISO/IEC 21000-5:2004. Some of these constraints are specified in various clauses and subclauses throughout ISO/IEC 21000-5:2004 and many are specified in Annex A (REL Schemas) of ISO/IEC 21000-5:2004. The text in 4.1.2 through 4.1.5 was copied from ISO/IEC 21000-5:2004. The tables in 4.1.2 through 4.1.5 were tabulated from the information in ISO/IEC 21000-5:2004. In the event of discrepancy between 4.1.2 through 4.1.5 and ISO/IEC 21000-5:2004, ISO/IEC 21000-5:2004 is authoritative.

NOTE The License Validation Rules Checker Reference Software in ISO/IEC 21000-8:—¹⁾ checks if an expression that only uses terms from ISO/IEC 21000-5:2004 satisfies all the expression constraints given in ISO/IEC 21000-5:2004.

7.1.2 Rules extracted from subclause 6.3 of ISO/IEC 21000-5:2004

The following provisions apply to `r:License` elements that contain `r:LicensePart` elements having the `r:licensePartId` or `r:licensePartIdRef` attributes.

1) To be published.

- a) **R-REL-2100:** An `r:LicensePart` shall not have both the `r:licensePartId` attribute and the `r:licensePartIdRef` attribute.
- b) **R-REL-2200:** For a given `r:LicensePartId` value v and `r:License` l , l shall not contain more than one `r:LicensePart` having an `r:licensePartId` attribute with the value v .
- c) **R-REL-2300:** If an `r:LicensePart` has an `r:licensePartIdRef` attribute, then it shall have empty content. As a corollary, types that are derivations of the type `r:LicensePart` should allow their content to be empty (for otherwise their `r:licensePartIdRef` attribute shall not be used).
- d) If an `r:LicensePart` a has an `r:licensePartIdRef` attribute with a certain value v , then there shall exist some (other) `r:LicensePart` b in the same `r:License` as a such that all of the following are true:
 - 1) **R-REL-2401:** b has an `r:licensePartId` attribute with value v ,
 - 2) **R-REL-2402:** the expanded element name of b exactly matches that of a , and
 - 3) **R-REL-2403:** b is not an ancestor of a .
- e) With the exception of signature verification and License issuer resolution (see 6.4), the following two processing steps shall be carried out before the other License processing steps defined in this part of ISO/IEC 21000. In particular, they shall be carried out before the evaluation of Variable references or the testing of equality.
 - 1) If an `r:LicensePart` a has an `r:licensePartIdRef` attribute with a certain value v , and b is the `r:LicensePart` in the same `r:License` as a that has an `r:licensePartId` attribute with value v , then the semantics of the `r:License` containing a and b are as if
 - i) a were removed from the `r:License` and replaced with an element α that is Equal to b ,
 - ii) the `r:licensePartId` attributes were removed from α and all of its descendants, and
 - iii) any preserved attributes that α has were removed from α , and any preserved attributes that a has were added to α , where here a preserved attribute is
 - I) any attribute of type `xsd:ID` or
 - II) any attribute for which `id` is the local part of its Qualified Name.
 - 2) If an `r:License` contains no `r:LicensePart` elements with an `r:licensePartIdRef` attribute, then the semantics of that `r:License` are as if the `r:licensePartId` attribute were removed from all `r:LicensePart` elements in that `r:License` with an `r:licensePartId` attribute.

NOTE 1 It is the intent of e)II) to enable the useful definition and incorporation of other identification systems on `r:LicensePart` elements beyond the document-global `xsd:ID`-typed identifiers.

NOTE 2 Circular references such as that shown in the first example below are not allowed because, after one application of provision e), the semantics will be as shown in the second example below, which violates provision d)3).

```
<r:license>
  <r:grantGroup licensePartId="x">
    <r:grantGroup licensePartIdRef="y"/>
  </r:grantGroup>
  <r:grantGroup licensePartId="y">
    <r:grantGroup licensePartIdRef="x"/>
  </r:grantGroup>
</r:license>
```

```

</r:grantGroup>
</r:license>

```

```

<r:license>
  <r:grantGroup licensePartId="x">
    <r:grantGroup>
      <r:grantGroup licensePartIdRef="x"/>
    </r:grantGroup>
  </r:grantGroup>
  <r:grantGroup licensePartId="y">
    <r:grantGroup>
      <r:grantGroup licensePartIdRef="y"/>
    </r:grantGroup>
  </r:grantGroup>
</r:license>

```

7.1.3 Rules extracted from subclauses 6.5.2 and 6.6 of ISO/IEC 21000-5:2004

Let *a* be an *r:LicensePart*. If *a/@r:varRef* exists, then **R-REL-3101**: *a* shall have empty content and **R-REL-3102**: the value of *a/@r:varRef* shall be the name of some Variable whose scope includes *a*.

R-REL-3200: If a conceptually abstract element appears in an *r:License*, it shall either have a type that is not conceptually abstract or appear in the form of a Variable reference, as described in 6.5.2.

R-REL-3200 (continued): If a conceptually abstract type appears in an *r:License*, it shall belong to an element that either is not conceptually abstract or appears in the form of a Variable reference, as described in 6.5.2.

The following is a list of the conceptually abstract elements and types.

Conceptually abstract elements	Conceptually abstract types
<i>r:licensePart</i>	<i>r:LicensePart</i>
<i>r:principal</i>	<i>r:Principal</i>
<i>r:right</i>	<i>r:Right</i>
<i>r:resource</i>	<i>r:Resource</i>
<i>r:condition</i>	<i>r:Condition</i>
<i>r:anXmlPatternAbstract</i>	<i>r:AnXmlPatternAbstract</i>
<i>r:principalPatternAbstract</i>	<i>r:PrincipalPatternAbstract</i>
<i>r:rightPatternAbstract</i>	<i>r:RightPatternAbstract</i>
<i>r:resourcePatternAbstract</i>	<i>r:ResourcePatternAbstract</i>
<i>r:conditionPatternAbstract</i>	<i>r:ConditionPatternAbstract</i>
<i>r:dcConstraint</i>	<i>r:DcConstraint</i>

r:trustRoot	r:TrustRoot
r:serviceDescription	r:ServiceDescription
r:propertyAbstract	r:PropertyAbstract
sx:name	sx:Name
	sx:StatefulCondition

7.1.4 Rules extracted from subclause 7.4.1 of ISO/IEC 21000-5:2004

R-REL-4100: If the syntax of a particular derivation of the type `r:LicensePart` declares its content model as optional, the semantics of that optional content model, unless specified otherwise, is that it shall not be omitted in an `r:LicensePart` of that derivation unless that `r:LicensePart` has an `r:licensePartIdRef` or `r:varRef` attribute.

The following is a list of types that cannot be empty unless used with `r:licensePartIdRef` or `r:varRef`.

Types	Elements using those types directly
r:Datum	r:datum
r:DepthConstraint	r:depthConstraint
r:DigitalResource	r:digitalResource and sx:WsdAddress/sx:kind/sx:wsdl and sx:WsdComplete/sx:wsdl
r:ExerciseMechanism	r:exerciseMechanism
r:ExistsRight	r:existsRight
r:Fulfiller	r:fulfiller
r:Grant	r:grant
r:GrantGroup	r:grantGroup
r:GrantGroupPattern	r:grantGroupPattern
r:GrantPattern	r:grantPattern
r:KeyHolder	r:keyHolder
r:PatternFromLicensePart	r:patternFromLicensePart
r:PrerequisiteRight	r:prerequisiteRight
r:PropertyPossessor	r:propertyPossessor
r:Revocable	r:revocable
r:RevocationFreshness	r:revocationFreshness
r:ServiceReference	r:serviceReference

r:TrustedRootGrants	r:trustedRootGrants
r:TrustedRootIssuers	r:trustedRootIssuers
sx:CallForCondition	sx:callForCondition
sx:FeeFlat	sx:feeFlat
sx:FeeMetered	sx:feeMetered
sx:FeePerInterval	sx:feePerInterval
sx:FeePerUse	sx:feePerUse
sx:FeePerUsePrePay	sx:feePerUsePrePay
sx:Rate	sx:rate
sx:StateReferenceValuePattern	sx:stateReferenceValuePattern
sx:Territory	sx:territory
sx:Uddi	sx:uddi
sx:ValidityIntervalDurationPattern	sx:validityIntervalDurationPattern
sx:ValidityIntervalStartsNow	sx:validityIntervalStartsNow
sx:ValidityTimePeriodic	sx:validityTimePeriodic
sx:WsdAddress	sx:wsdlAddress
sx:WsdComplete	sx:wsdlComplete
anonymous type defined in mx:complement	mx:complement
mx:Destination	mx:destination
mx:DiCriteria	mx:diCriteria
mx:DiItemReference	mx:diItemReference
mx:DiPartOf	mx:diPartOf
mx:DiReference	mx:diReference
mx:IsMarked	mx:isMarked
mx:Mark	mx:mark
mx:ResourceSignedBy	mx:resourceSignedBy
mx:Source	mx:source
mx:Transaction	mx:transaction

The following is a list of types that declare an optional content model but specify an *exception* to the rule of 7.4.1.

Types	Elements using those types directly
anonymous type defined in <code>mx:union</code>	<code>mx:union</code>
anonymous type defined in <code>mx:intersection</code>	<code>mx:intersection</code>
<code>mx:RequiredAttributeChanges</code>	<code>mx:requiredAttributeChanges</code>
<code>mx:ProhibitedAttributeChanges</code>	<code>mx:prohibitedAttributeChanges</code>

7.1.5 Rules extracted from subclause 7.3.6 of ISO/IEC 21000-5:2004

R-REL-5100: An `r:EncryptedContent` shall have an `xenc:Type` attribute with a value of `http://www.w3.org/2001/04/xmlenc#Content`.

7.2 REL Parser Conformance

An REL Parser is defined as a module that takes a file as input and produces as output either 1 (if the input file is a conformant Rights Expression) or 0 (if the input file is not a conformant Rights Expression).

An REL Parser is conformant only if, given the inputs given in Annex E it produces the corresponding outputs from Annex E.

8 REL Amd.1 Conformance

8.1 Introduction

In Annex F, test streams are defined for ISO/IEC 21000-5:2004/Amd.1:2007.

8.2 Rules extracted from subclause 6.3 of ISO/IEC 21000-5:2004

The following provisions apply to `r:License` elements that contain `r:LicensePart` elements having the `r:licensePartId` or `r:licensePartIdRef` attributes.

- a) **R-REL-2100:** An `r:LicensePart` shall not have both the `r:licensePartId` attribute and the `r:licensePartIdRef` attribute.
- b) **R-REL-2200:** For a given `r:LicensePartId` value *v* and `r:License` *l*, *l* shall not contain more than one `r:LicensePart` having an `r:licensePartId` attribute with the value *v*.
- c) **R-REL-2300:** If an `r:LicensePart` has an `r:licensePartIdRef` attribute, then it shall have empty content. As a corollary, types that are derivations of the type `r:LicensePart` should allow their content to be empty (for otherwise their `r:licensePartIdRef` attribute shall not be used).
- d) If an `r:LicensePart` *a* has an `r:licensePartIdRef` attribute with a certain value *v*, then there shall exist some (other) `r:LicensePart` *b* in the same `r:License` as *a* such that all of the following are true:
 1. **R-REL-2401:** *b* has an `r:licensePartId` attribute with value *v*,
 2. **R-REL-2402:** the expanded element name of *b* exactly matches that of *a*, and
 3. **R-REL-2403:** *b* is not an ancestor of *a*.

- e) With the exception of signature verification and License issuer resolution (see 6.4), the following two processing steps shall be carried out before the other License processing steps defined in this part of ISO/IEC 21000. In particular, they shall be carried out before the evaluation of Variable references or the testing of equality.
1. If an `r:LicensePart` *a* has an `r:licensePartIdRef` attribute with a certain value *v*, and *b* is the `r:LicensePart` in the same `r:License` as *a* that has an `r:licensePartId` attribute with value *v*, then the semantics of the `r:License` containing *a* and *b* are as if
 - i) *a* were removed from the `r:License` and replaced with an element α that is Equal to *b*,
 - ii) the `r:licensePartId` attributes were removed from α and all of its descendants, and
 - iii) any preserved attributes that α has were removed from α , and any preserved attributes that *a* has were added to α , where here a preserved attribute is
 - I) any attribute of type `xsd:ID` or
 - II) any attribute for which `id` is the local part of its Qualified Name.
 2. If an `r:License` contains no `r:LicensePart` elements with an `r:licensePartIdRef` attribute, then the semantics of that `r:License` are as if the `r:licensePartId` attribute were removed from all `r:LicensePart` elements in that `r:License` with an `r:licensePartId` attribute.

NOTE 1 It is the intent to enable the useful definition and incorporation of other identification systems on `r:LicensePart` elements beyond the document-global `xsd:ID`-typed identifiers.

8.3 Rules extracted from subclauses 6.5.2 and 6.6 of ISO/IEC 21000-5:2004

Let *a* be an `r:LicensePart`. If *a*/`@r:varRef` exists, then **R-REL-3101**: *a* shall have empty content and **R-REL-3102**: the value of *a*/`@r:varRef` shall be the name of some Variable whose scope includes *a*.

R-REL-3200: If a conceptually abstract element appears in an `r:License`, it shall either have a type that is not conceptually abstract or appear in the form of a Variable reference, as described in 6.5.2.

R-REL-3200 (continued): If a conceptually abstract type appears in an `r:License`, it shall belong to an element that either is not conceptually abstract or appears in the form of a Variable reference, as described in 6.5.2.

The following is the list of the conceptually abstract elements and types.

Conceptually abstract elements	Conceptually abstract types
<code>r:licensePart</code>	<code>r:LicensePart</code>
<code>r:principal</code>	<code>r:Principal</code>
<code>r:right</code>	<code>r:Right</code>
<code>r:resource</code>	<code>r:Resource</code>
<code>r:condition</code>	<code>r:Condition</code>
<code>r:trustRoot</code>	<code>r:TrustRoot</code>
<code>r:serviceDescription</code>	<code>r:ServiceDescription</code>
<code>r:propertyAbstract</code>	<code>r:PropertyAbstract</code>

8.4 Rules extracted from subclause 7.4.1 of ISO/IEC 21000-5:2004

R-REL-4100: If the syntax of a particular derivation of the type `r:LicensePart` declares its content model as optional, the semantics of that optional content model, unless specified otherwise, is that it shall not be omitted in an `r:LicensePart` of that derivation unless that `r:LicensePart` has an `r:licensePartIdRef` or `r:varRef` attribute.

The following is a list of types that cannot be empty unless used with `r:licensePartIdRef` or `r:varRef`.

Types	Elements using those types directly
<code>r:DigitalResource</code>	<code>r:digitalResource</code>
<code>r:Grant</code>	<code>r:grant</code>
<code>r:KeyHolder</code>	<code>r:keyHolder</code>
<code>r:PropertyPossessor</code>	<code>r:propertyPossessor</code>
<code>r:ServiceReference</code>	<code>r:serviceReference</code>
<code>r:TrustedRootIssuers</code>	<code>r:trustedRootIssuers</code>
<code>sx:Territory</code>	<code>sx:territory</code>

8.5 Rules for the DrmSystem condition

R-REL-5100: If a license has the `mlx:drmSystem` condition, then the right element of the license shall be `mlx:GovernedCopy` or `mlx:GovernedMove`.

9 REL Amd.2 Conformance

9.1 Introduction

The MPEG-21 REL DAC Profile validation rules defined for the elements of the MPEG-21 REL DAC Profile are a subset of the rules previously defined in MPEG-21 REL Validation Rules for the complete MPEG-21 REL standard specification.

For each of the rules, examples are provided to show how the rule can be violated.

In Annex G, test streams are defined for ISO/IEC 21000-5:2004/Amd.2:2007.

9.2 LicensePartId and LicensePartIdRef

The Rule

These rules are given in 6.3 of the REL. The following provisions apply to `r:License` elements that contain `r:LicensePart` elements having the `r:licensePartId` or `r:licensePartIdRef` attributes.

- An `r:LicensePart` shall not have both the `r:licensePartId` attribute and the `r:licensePartIdRef` attribute.
- For a given `r:LicensePartId` value *v* and `r:License` *l*, *l* shall not contain more than one `r:LicensePart` having an `r:licensePartId` attribute with the value *v*.

- c) If an `r:LicensePart` has an `r:licensePartIdRef` attribute, then it shall have empty content. As a corollary, types that are derivations of the type `r:LicensePart` should allow their content to be empty (for otherwise their `r:licensePartIdRef` attribute shall not be used).
- d) If an `r:LicensePart` *a* has an `r:licensePartIdRef` attribute with a certain value *v*, then there shall exist some (other) `r:LicensePart` *b* in the same `r:License` as *a* such that all of the following are true:
- 1) *b* has an `r:licensePartId` attribute with value *v*,
 - 2) the expanded element name of *b* exactly matches that of *a*, and
 - 3) *b* is not an ancestor of *a*.
- e) With the exception of signature verification and License issuer resolution (see 6.4), the following two processing steps shall be carried out before the other License processing steps defined in this part of ISO/IEC 21000. In particular, they shall be carried out before the evaluation of Variable references or the testing of equality.
- 1) If an `r:LicensePart` *a* has an `r:licensePartIdRef` attribute with a certain value *v*, and *b* is the `r:LicensePart` in the same `r:License` as *a* that has an `r:licensePartId` attribute with value *v*, then the semantics of the `r:License` containing *a* and *b* are as if
 - i) *a* were removed from the `r:License` and replaced with an element α that is Equal to *b*,
 - ii) the `r:licensePartId` attributes were removed from α and all of its descendants, and
 - iii) any preserved attributes that α has were removed from α , and any preserved attributes that *a* has were added to α , where here a preserved attribute is
 - I) any attribute of type `xsd:ID` or
 - II) any attribute for which `id` is the local part of its Qualified Name.
 - 2) If an `r:License` contains no `r:LicensePart` elements with an `r:licensePartIdRef` attribute, then the semantics of that `r:License` are as if the `r:licensePartId` attribute were removed from all `r:LicensePart` elements in that `r:License` with an `r:licensePartId` attribute.

NOTE 1 It is the intent to enable the useful definition and incorporation of other identification systems on `r:LicensePart` elements beyond the document-global `xsd:ID`-typed identifiers.

Examples that Violate the Rules

- a) An `r:LicensePart` shall not have both the `r:licensePartId` attribute and the `r:licensePartIdRef` attribute.

```
<r:license>
  <r:grant>
    <r:keyHolder licensePartId="Eva">
      <r:info>
        <dsig:KeyValue>
          <dsig:RSAKeyValue>
            <dsig:Modulus>KtdToQQyzA==</dsig:Modulus>
            <dsig:Exponent>AQABAA==</dsig:Exponent>
          </dsig:RSAKeyValue>
        </dsig:KeyValue>
      </r:info>
    </r:keyHolder>
    <mx:play/>
    <r:digitalResource>
      <r:nonSecureIndirect URI="http://www.onlinemusic.com/song.mp3"/>
    </r:digitalResource>
  </r:grant>
</r:license>
```

```

    </r:digitalResource>
    <r:validityInterval>
      <r:notBefore>2003-01-01T00:00:00</r:notBefore>
      <r:notAfter>2003-12-31T12:59:59</r:notAfter>
    </r:validityInterval>
  </r:grant>
  <r:grant>
    <r:keyHolder licensePartId="Silvia" licensePartIdRef="Eva"/>
    <mx:play/>
    <r:digitalResource>
      <r:nonSecureIndirect URI="http://www.onlinemusic.com/song.mp3"/>
    </r:digitalResource>
    <r:validityInterval>
      <r:notBefore>2003-01-01T00:00:00</r:notBefore>
      <r:notAfter>2003-12-31T12:59:59</r:notAfter>
    </r:validityInterval>
  </r:grant>
</r:license>

```

- b) For a given `r:LicensePartId` value v and `r:License` l , l shall not contain more than one `r:LicensePart` having an `r:licensePartId` attribute with the value v .

```

<r:license>
  <r:grant>
    <r:keyHolder licensePartId="Alice">
      <r:info>
        <dsig:KeyValue>
          <dsig:RSAKeyValue>
            <dsig:Modulus>KtdToQQyzA==</dsig:Modulus>
            <dsig:Exponent>AQABAA==</dsig:Exponent>
          </dsig:RSAKeyValue>
        </dsig:KeyValue>
      </r:info>
    </r:keyHolder>
    <mx:play/>
    <r:digitalResource licensePartId="Alice">
      <r:nonSecureIndirect URI="http://www.onlinemusic.com/song.mp3"/>
    </r:digitalResource>
    <r:validityInterval>
      <r:notBefore>2003-01-01T00:00:00</r:notBefore>
      <r:notAfter>2003-12-31T12:59:59</r:notAfter>
    </r:validityInterval>
  </r:grant>
</r:license>

```

- c) If an `r:LicensePart` has an `r:licensePartIdRef` attribute, then it shall have empty content.

```

<r:license>
  <r:grant>
    <r:keyHolder licensePartId="Alice">
      <r:info>
        <dsig:KeyValue>
          <dsig:RSAKeyValue>
            <dsig:Modulus>KtdToQQyzA==</dsig:Modulus>
            <dsig:Exponent>AQABAA==</dsig:Exponent>
          </dsig:RSAKeyValue>
        </dsig:KeyValue>
      </r:info>
    </r:keyHolder>
    <mx:play/>
    <r:digitalResource>
      <r:nonSecureIndirect URI="http://www.onlinemusic.com/song.mp3"/>
    </r:digitalResource>
    <r:validityInterval>

```

```

        <r:notBefore>2003-01-01T00:00:00</r:notBefore>
        <r:notAfter>2003-12-31T12:59:59</r:notAfter>
    </r:validityInterval>
</r:grant>
<r:grant>
    <r:keyHolder licensePartIdRef="Alice">
        <r:info>
            <dsig:KeyValue>
                <dsig:RSAKeyValue>
                    <dsig:Modulus>KtdToQQyzA==</dsig:Modulus>
                    <dsig:Exponent>AQABAA==</dsig:Exponent>
                </dsig:RSAKeyValue>
            </dsig:KeyValue>
        </r:info>
    </r:keyHolder>
    <r:digitalResource>
        <r:nonSecureIndirect URI="http://www.onlinemusic.com/song.mp3"/>
    </r:digitalResource>
    <r:validityInterval>
        <r:notBefore>2003-01-01T00:00:00</r:notBefore>
        <r:notAfter>2003-12-31T12:59:59</r:notAfter>
    </r:validityInterval>
</r:grant>
<r:issuer/>
</r:license>

```

d) If an `r:LicensePart` *a* has an `r:licensePartIdRef` attribute with a certain value *v*, then there shall exist some (other) `r:LicensePart` *b* in the same `r:License` as *a* such that all of the following are true:

- 1) *b* has an `r:licensePartId` attribute with value *v*,

```

<r:license>
    <r:grant>
        <r:keyHolder licensePartId="Alice">
            <r:info>
                <dsig:KeyValue>
                    <dsig:RSAKeyValue>
                        <dsig:Modulus>KtdToQQyzA==</dsig:Modulus>
                        <dsig:Exponent>AQABAA==</dsig:Exponent>
                    </dsig:RSAKeyValue>
                </dsig:KeyValue>
            </r:info>
        </r:keyHolder>
        <mx:play/>
        <r:digitalResource>
            <r:nonSecureIndirect URI="http://www.onlinemusic.com/song.mp3"/>
        </r:digitalResource>
        <r:validityInterval>
            <r:notBefore>2003-01-01T00:00:00</r:notBefore>
            <r:notAfter>2003-12-31T12:59:59</r:notAfter>
        </r:validityInterval>
    </r:grant>
    <r:grant>
        <r:keyHolder licensePartIdRef="Lora"/>
        <mx:enlarge/>
        <r:digitalResource>
            <r:nonSecureIndirect URI="http://www.onlinemusic.com/song.mp3"/>
        </r:digitalResource>
        <r:validityInterval>
            <r:notBefore>2003-01-01T00:00:00</r:notBefore>
            <r:notAfter>2003-12-31T12:59:59</r:notAfter>
        </r:validityInterval>
    </r:grant>
    <r:issuer/>
</r:license>

```

- 2) the expanded element name of *b* exactly matches that of *a*

```
<r:license>
  <r:grant>
    <r:keyHolder licensePartId="Alice">
      <r:info>
        <dsig:KeyValue>
          <dsig:RSAKeyValue>
            <dsig:Modulus>KtdToQQyzA==</dsig:Modulus>
            <dsig:Exponent>AQABAA==</dsig:Exponent>
          </dsig:RSAKeyValue>
        </dsig:KeyValue>
      </r:info>
    </r:keyHolder>
    <mx:play/>
    <r:digitalResource licensePartIdRef="Alice"/>
    <r:validityInterval>
      <r:notBefore>2003-01-01T00:00:00</r:notBefore>
      <r:notAfter>2003-12-31T12:59:59</r:notAfter>
    </r:validityInterval>
  </r:grant>
  <r:issuer/>
</r:license>
```

- 3) *b* is not an ancestor of *a*.

A direct example:

```
<r:license>
  <r:grant licensePartId="x">
    <r:issue/>
    <r:grant licensePartIdRef="x"/>
  </r:grant>
</r:license>
```

9.3 Variable reference and conceptually abstract elements and types

The Rules

6.5.2 and 6.6 of the REL specify the following rules:

Let *a* be an `r:LicensePart`. If `a/@r:varRef` exists, then *a* shall have empty content and the value of `a/@r:varRef` shall be the name of some Variable whose scope includes *a*.

If a conceptually abstract element appears in an `r:License`, it shall either have a type that is not conceptually abstract or appear in the form of a Variable reference, as described in 6.5.2.

If a conceptually abstract type appears in an `r:License`, it shall belong to an element that either is not conceptually abstract or appears in the form of a Variable reference, as described in 6.5.2.

The following is a list of the conceptually abstract elements and types.

Conceptually abstract elements	Conceptually abstract types
r:licensePart	r:LicensePart
r:principal	r:Principal
r:right	r:Right
r:resource	r:Resource
r:condition	r:Condition
r:trustRoot	r:TrustRoot
r:serviceDescription	r:ServiceDescription
r:propertyAbstract	r:PropertyAbstract

Examples that Violate the Rules

Let *a* be an r:LicensePart. If *a/@r:varRef* exists, then *a* shall have empty content

```
<r:license>
  <r:grant>
    <r:forAll varName="AliceClub">
      <r:propertyPossessor>
        <sx:propertyUri definition="urn:edu:elibrary"/>
      </r:propertyPossessor>
    </r:forAll>
    <r:principal varRef="AliceClub">
      <r:keyHolder/>
    </r:principal>
    <mx:print/>
    <r:digitalResource>
      <r:nonSecureIndirect URI="http://www.onlinemusic.com/mySong.mp3"/>
    </r:digitalResource>
    <r:validityInterval>
      <r:notBefore>2003-01-01T00:00:00</r:notBefore>
      <r:notAfter>2003-12-31T12:59:59</r:notAfter>
    </r:validityInterval>
  </r:grant>
  <r:issuer/>
</r:license>
```

and the value of *a/@r:varRef* shall be the name of some Variable whose scope includes *a*.

```
<r:license>
  <r:grant>
    <r:forAll varName="AliceClub">
      <r:propertyPossessor>
        <sx:propertyUri definition="urn:edu:elibrary"/>
      </r:propertyPossessor>
    </r:forAll>
    <r:principal varRef="AliceClub1"/>
    <mx:print/>
    <r:digitalResource>
      <r:nonSecureIndirect URI="http://www.onlinemusic.com/mySong.mp3"/>
    </r:digitalResource>
    <r:validityInterval>
```

```

        <r:notBefore>2003-01-01T00:00:00</r:notBefore>
        <r:notAfter>2003-12-31T12:59:59</r:notAfter>
    </r:validityInterval>
</r:grant>
<r:issuer/>
</r:license>

```

If a conceptually abstract element appears in an `r:License`, it shall either have a type that is not conceptually abstract or appear in the form of a Variable reference, as described in 6.5.2.

An example of an element of a conceptually abstract type but with no variable reference:

```

<r:license>
  <r:grant>
    <r:keyHolder>
      <r:info>
        <dsig:KeyValue>
          <dsig:RSAKeyValue>
            <dsig:Modulus>KtdToQQyzA==</dsig:Modulus>
            <dsig:Exponent>AQABAA==</dsig:Exponent>
          </dsig:RSAKeyValue>
        </dsig:KeyValue>
      </r:info>
    </r:keyHolder>
    <r:right xsi:type="r:Right"/>
    <mx:diReference>
      <mx:identifier>someResourceUri</mx:identifier>
    </mx:diReference>
  </r:grant>
<r:issuer/>
</r:license>

```

Another example of an element of a conceptually abstract type but with no variable reference:

```

<r:license>
  <r:grant>
    <r:keyHolder>
      <r:info>
        <dsig:KeyValue>
          <dsig:RSAKeyValue>
            <dsig:Modulus>KtdToQQyzA==</dsig:Modulus>
            <dsig:Exponent>AQABAA==</dsig:Exponent>
          </dsig:RSAKeyValue>
        </dsig:KeyValue>
      </r:info>
    </r:keyHolder>
    <r:right/>
    <mx:diReference>
      <mx:identifier>someResourceUri</mx:identifier>
    </mx:diReference>
  </r:grant>
<r:issuer/>
</r:license>

```

9.4 LicensePart optional content model

The Rules

7.4.1 of the REL specifies the following rule:

If the syntax of a particular derivation of the type `r:LicensePart` declares its content model as optional, the semantics of that optional content model, unless specified otherwise, is that it shall not be omitted in an `r:LicensePart` of that derivation unless that `r:LicensePart` has an `r:licensePartIdRef` or `r:varRef` attribute.

The following is a list of types that cannot be empty unless used with `r:licensePartIdRef` or `r:varRef`.

Types	Elements using those types directly
<code>r:DigitalResource</code>	<code>r:digitalResource</code>
<code>r:Grant</code>	<code>r:grant</code>
<code>r:KeyHolder</code>	<code>r:keyHolder</code>
<code>r:PropertyPossessor</code>	<code>r:propertyPossessor</code>
<code>r:ServiceReference</code>	<code>r:serviceReference</code>
<code>r:TrustedRootIssuers</code>	<code>r:trustedRootIssuers</code>
<code>sx:Territory</code>	<code>sx:territory</code>

Examples that Violate the Rules

```
<r:license>
  <r:grant>
    <r:keyHolder/>
    <mx:play/>
    <r:digitalResource>
      <r:nonSecureIndirect URI="http://www.onlinemusic.com/mySong.mp3"/>
    </r:digitalResource>
  </r:grant>
  <r:issuer/>
</r:license>
```

10 RDD Conformance

RDD Conformance is not defined at this time because there is no syntax.

11 DIA Conformance

11.1 DIA Description Conformance

A DIA Description is conformant only if it is well-formed XML and meets all the description constraints specified in ISO/IEC 21000-7:2004. These constraints are specified in text and schemas in various clauses and subclauses throughout ISO/IEC 21000-7:2004.

In particular, when the DIA Description is a Bitstream Syntax Description (BSD), the following additional constraint applies: a BSD is conformant only if it is BSD-1 valid, as specified in subclause 8.3.1.5 of ISO/IEC 21000-7:2004.

11.2 Bitstream Syntax Schema Conformance

A Bitstream Syntax Schema (BS Schema) is conformant only if it is BSD-2 valid, as specified in subclause 8.4.6 of ISO/IEC 21000-7:2004.

11.3 DIA Parser Conformance

A DIA Parser is defined as a module that takes a file as input and produces as output either 1 (if the input file is a conformant DIA Description) or 0 (if the input file is not a conformant DIA Description).

A DIA Parser is conformant only if, given the inputs given in Annex H it produces the corresponding outputs from Annex H.

11.4 DIA BintotBSD Parser Conformance

A **BintotBSD Parser** is defined as a processor that takes a BS Schema and a Bitstream as inputs and generates a BS Description describing the Bitstream.

The conformance testing of a BintotBSD Parser involves the comparison of the result of generating a BS Description using the reference BintotBSD Parser against the result using the test BintotBSD. The conformance testing checks whether the generated BS Descriptions are canonically equal. Canonical is understood as in the "Canonical XML" Recommendation.

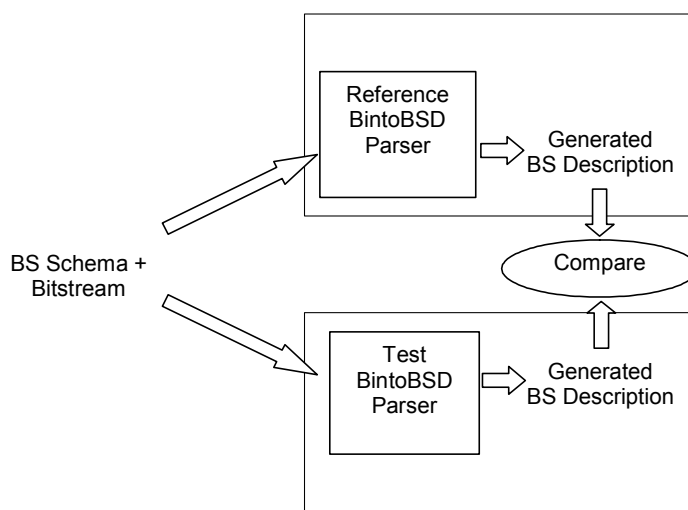


Figure 1 — Conformance testing of the BintotBSD Parser

11.5 DIA BSDtoBin Parser Conformance

A **BSDtoBin Parser** is defined as a processor that takes a BS Description, the corresponding BS Schema and the described Bitstream as inputs and generates a new Bitstream.

The conformance testing of a BSDtoBin Parser involves the comparison of the result of generating a Bitstream using the reference BSDtoBin Parser against the result using the test BSDtoBin. The conformance testing checks whether the generated Bitstreams are equal bit-to-bit.

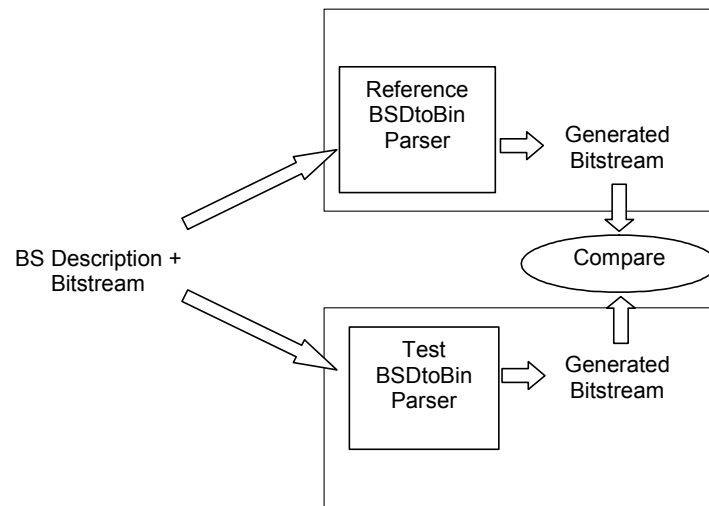


Figure 2 — Conformance testing of the BSDtoBin Parser

11.6 DIA gBSDtoBin Parser Conformance

A **gBSDtoBin Parser** is defined as a processor that takes a gBS Description and the described Bitstream as inputs and generates a new Bitstream.

The conformance testing of a gBSDtoBin Parser involves the comparison of the result of generating a Bitstream using the reference gBSDtoBin Parser against the result using the test gBSDtoBin. The conformance testing checks whether the generated Bitstreams are equal bit-to-bit.

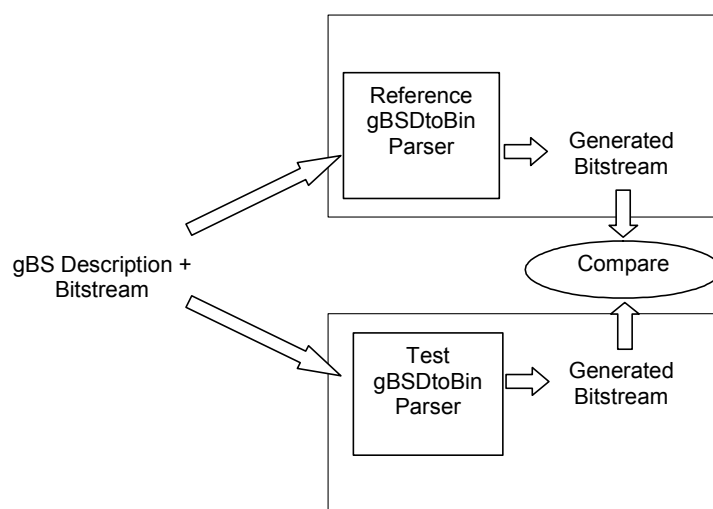


Figure 3 — Conformance testing of the gBSDtoBin Parser

11.7 DIA gBSDtoAU Conformance

Annex H describes four XML bitstreams to be used for conformance testing of DIA gBSDtoAU.

12 DIA Amd.1 Conformance

Annex I comprises conformance bitstreams for ISO/IEC 21000-7:2004/Amd.1:2006 DIA conversions and permissions including appropriate descriptions.

13 DIA Amd.2 Conformance

13.1 Simple_Mode_XML_Fragmenter

Annex J describes conformance streams consisting of gBSDs for different types of media which are enriched by XML streaming instructions properties (as XML attributes). Given as input to the fragmenter process, they shall be fragmented into PUs according to ISO/IEC 21000-7:2004/Amd.2:2007.

14 File Format Conformance

14.1 Introduction

This section describes the conformance suite for the file format standard in ISO/IEC 21000-9:2005.

The purpose of the conformance suite is to cover the set of valid features that may be exercised in the file format. Media conformance is not covered, though of course in order to exercise the file format features, media has to be stored.

In order to assure coverage of features, the associated spreadsheet is used to list the features in groups, and to document which files exercise each feature. (Features not currently covered by any file are marked with the sign “---“.)

Note that ‘multi-function’ files that include both MPEG-21 elements and elements from other standards (including MPEG-4) can be created; this conformance suite does not cover such ‘hybrid’ files.

14.2 Process

Those wishing to check the conformance of their implementation should perform the following checks. First, all conformance should check the “basic box handling” section of the tests, as this is common to all specifications. Then, the mandatory features of the selected specification should be checked, and finally, of course, those optional features that the implementation being checked also covers.

The suite of conformance tests does not currently cover deliberately errored files. However, such files do occur in practice and implementations should be written to be resilient.

There is no tool provided to check the conformance of files. However, such tools do exist; the reference software can be used to open files in ‘debug’ mode and provide a listing of what it finds, and other trade associations and standards bodies may have validation tools tailored to their areas.

14.3 Areas tested

Annex K specifies an Excel document that has two spreadsheets. The first briefly lists the areas and features covered and then has a column for each proposed file. The second sheet provides a brief description of each area and feature, by line.

14.4 File Documentation

Annex K specifies the file documentation.

15 DIP Conformance

15.1 Introduction

For ISO/IEC 21000-10:2006 the following general conformance points are defined:

- DIP schema conformance;
- DIP engine conformance;
- DIML syntax conformance;
- DIM engine conformance; and
- J-DIXO language conformance.

The remaining subclauses of this clause provide further detail on these conformance points.

15.2 DIP schema conformance

15.2.1 Introduction

ISO/IEC 21000-10:2006 specifies several W3C XML Schema based definitions for XML elements (referred to as DIP elements in this clause) and types that enable

- construction of an Object Map;
- declaration of Digital Item Methods (DIMs) in a Digital Item; and
- declaration of Java classes implementing J-DIXOs.

DIP schema conformance refers to the validity of any such DIP element against the XML Schema definitions specified in ISO/IEC 21000-10:2006.

15.2.2 Testing DIP schema conformance

A DIP element is DIP schema conformant if it is valid against the XML schema definition of that element as specified in ISO/IEC 21000-10:2006.

Such conformance can be tested by parsing an XML document containing as document element the DIP element with the reference DIP Schema Parser. This parser takes as input such an XML document and outputs either 1 (the document is conformant to the DIP schema) or 0 (the document is not conformant to the DIP schema).

NOTE Typically DIP elements will not appear on their own in an XML document but will be included within a DIDL document. In this case the DIP elements would need to be extracted first and then tested. However, for convenience, the reference DIP Schema Parser also provides the capability of accepting a DIDL document and testing all contained DIP elements for conformance to the DIP schema.

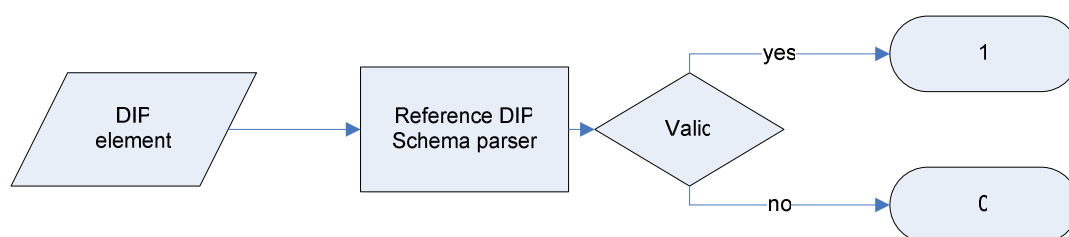


Figure 4 — Testing DIP schema conformance of a DIP element

EXAMPLE 1 Example of a valid DIP MethodInfo element

```
<MethodInfo xmlns="urn:mpeg:mpeg21:2005:01-DIP-NS">
  <Argument>http://www.foo.com/objecttype</Argument>
</MethodInfo>
```

NOTE 1 The value `http://www.foo.com/objecttype` in the above example is an example value only.

EXAMPLE 2 Example of a valid DIP Label element

```
<Label xmlns="urn:mpeg:mpeg21:2005:01-DIP-NS">urn:mpeg:mpeg21:2005:01-DIP-NS:DIM</Label>
```

EXAMPLE 3 Example of a valid DIP ObjectType element

```
<ObjectType xmlns="urn:mpeg:mpeg21:2005:01-DIP-NS">
  http://www.foo.com/objecttype
</ObjectType>
```

NOTE 2 The value `http://www.foo.com/objecttype` in the above example is an example value only.

EXAMPLE 4 Example of a valid DIP JDIXOClasses element

```
<JDIXOClasses xmlns="urn:mpeg:mpeg21:2005:01-DIP-NS">
  <Class>com.foo.TestJDIXO</Class>
</JDIXOClasses>
```

NOTE 3 The value `com.foo.TestJDIXO` in the above example is an example value only.

15.2.3 DIP engine conformance

In Annex L, test DIDs are defined for the different DIP conformance points.

15.2.4 Introduction

The DIP elements are not intended to be used in isolation but to be contained within a DIDL document to enable the DIP processing of a Digital Item. Such processing includes actions such as recognizing processable DIM declarations, recognizing Object Map related information, and making DIP related information (e.g. the available DIMs that can be executed, and/or the available Objects that can be operated on) available to the User and DIM author (within the DIM definitions via calls to certain DIBOs).

15.2.5 Testing DIP engine conformance

Such processing can be tested by appropriately authored DIDL documents and DIMs and checking the results of a DIP engine against the expected results.

The reference DIP engine in conjunction with the test cases specified in L.1 can be used to test a DIP engine for such DIP engine conformance.

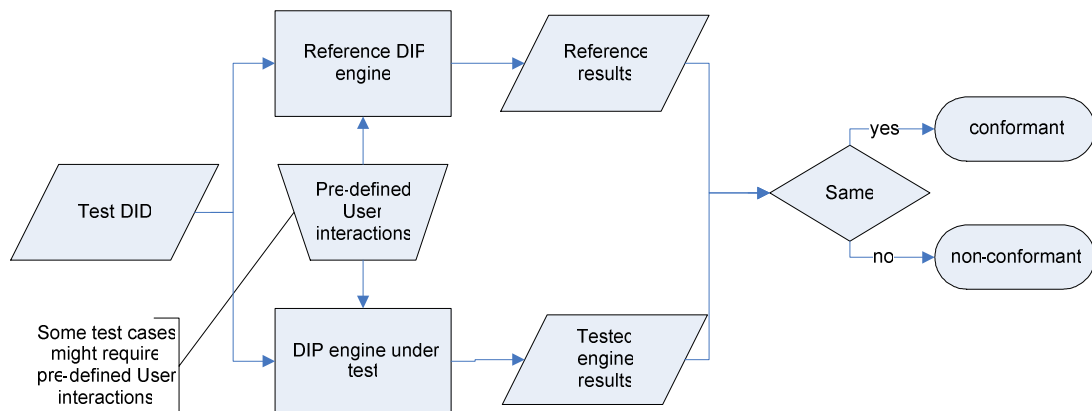


Figure 5 — Testing DIP engine conformance

The DIP elements in all Test DIDs shall be checked for DIP schema conformance prior to release for use in DIP engine conformance testing.

15.3 DIML syntax conformance

15.3.1 Introduction

DIM definitions are authored in DIML which is an extension of ECMAScript including the DIML global and local objects, and the normative set of DIBOs as specified in ISO/IEC 21000-10:2006.

DIML syntax conformance refers to the syntactic correctness of a DIM definition against the syntax of DIML as specified in ISO/IEC 21000-10:2006.

15.3.2 Testing DIML syntax conformance

A DIM definition is DIML syntax conformant if it is syntactically valid according to the DIML as specified in ISO/IEC 21000-10:2006.

Such conformance can be tested by running the DIM definition with the reference DIP engine, which includes parsing the DIML of the DIM definition as part of the execution of the DIM. If the DIM definition is not DIML syntax conformant, the reference DIP engine will indicate a syntax error in the DIML.

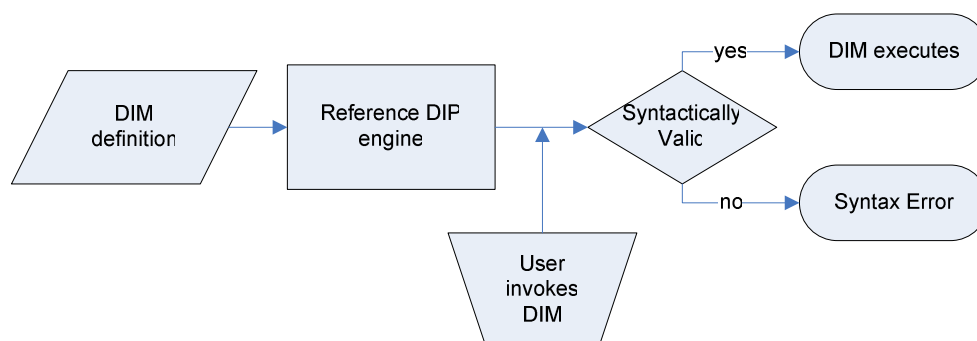


Figure 6 — Testing DIML syntax conformance of a DIM definition

NOTE In practice a valid DIDL containing a DIM declaration for the DIM definition is passed as input to the reference DIP engine.

EXAMPLE Example of a valid DIM definition

```
function helloWorld()
{
    DIP.alert( "Hello world", MSG_INFO);
}
```

15.4 DIM engine conformance

15.4.1 Introduction

The DIM definitions are intended to provide suggested User interactions with the Digital Item as suggested by the DIM author. Such interactions are the result of processing the DIML of the DIM definition. Such processing includes execution of the DIBOs called by the DIM, and hence implementation of the normative semantics of the DIBOs. That is, DIBO implementation conformance can be subsumed into DIM engine conformance.

15.4.2 DIBOs with non-deterministic results

15.4.2.1 Introduction

For some DIBOs, the results of the implementation of the normative semantics of the DIBO can result in non-deterministic results due to the nature of the semantics of the DIBO. DIBOs related to RDD verbs are an example of such DIBOs (e.g. DIP.play), as are DIBOs requiring direct interaction with the User (e.g. DIP.alert).

In such cases conformance points will be stated in terms that will be to some extent subjective in nature.

The DIBOs in this category are discussed in the following subclauses.

15.4.2.2 DIA.adapt

The theoretical number of adaptations that could be invoked by calling DIA.adapt is unbounded. In addition, according to the semantics, an implementation can return a null if it does not support the requested adaptation. Further, DIP itself does not specify any requirements in regards to the adaptation process itself, the results of the adaptation, or about information used to drive the adaptation. From this perspective, the functionality of the DIA.adapt DIBO is to provide the normative interface by which a DIM author can request an adaptation (that might or might not be supported) by the underlying MPEG-21 environment.

Thus for DIA.adapt, test cases for a finite set of example DIA based adaptations are provided where the adaptation results generated by two DIM engines can be compared.

NOTE While the adaptation results can be compared, the equivalence of the adaptation results is not actually part of the DIP conformance, since that is outside the scope of DIP.

In the actual conformance testing process, a DIM engine that does not support a particular adaptation from the example set shall return a null value, as per the semantics of DIA.adapt. A DIM engine that does support a particular adaptation shall return the adapted element.

Since the actual adapted element could be implementation dependent (for example one implementation might create a temporary file storing the associated adapted resource, another implementation might have a reference to a pre-adapted resource on a server, etc) the confirmation of the adaptation result is determined by the tester.

15.4.2.3 DID.configureChoice

The DID.configureChoice DIBO allows a User to configure a given Choice. The results of such configuration are dependent on the selections made by the User. Thus test cases for DID.configureChoice will also specify the pre-defined User interactions to configure a Choice in order to produce known results.

In addition, the implementer of DID.configureChoice may allow the User configuration of the Choice to be done in a manner appropriate to the MPEG-21 environment in which the DIM engine is executing. The conformance testing is testing only the correct configuration of the Choice according to the User interactions, not the manner of such configuration.

15.4.2.4 DIP.alert

The DIP.alert DIBO provides simple textual feedback to the User. The implementer of the DIBO may provide this feedback in a way that is appropriate to the MPEG-21 environment in which the DIM engine is executing.

EXAMPLE A server based process might log the message to a file, while a desktop application might pop-up a window in a GUI.

Thus the manner of providing the textual feedback to the User may vary between implementations. The core semantics of the DIBO is that the feedback is provided.

In the reference DIM engine a pop-up alert window is provided. For other DIM engines being tested, a subjective decision by the tester could be required as to whether the semantics of the DIP.alert DIBO have been implemented.

15.4.2.5 DIP.execute

The DIP.execute DIBO allows a DIM author to request an executable resource to be executed. If the implementation is able to initiate the execution of the resource it returns true, otherwise it returns false. DIP itself does not place any requirements on the executable resource or its execution. From this perspective, the functionality of the DIP.execute DIBO is to provide the normative interface by which a DIM author can request the execution of an executable resource (that might or might not be initiated) by the underlying MPEG-21 environment.

NOTE The execution of the executable resource might not be initiated for a variety of reasons. For example the type of executable resource might not be executable on a given platform. Or the execution might not be allowed due to some security restrictions.

For DIP.execute a test case with a pre-compiled executable for Windows platform is provided.

In the actual conformance testing process, a DIM engine that does not support or allow execution of such an executable resource shall return a false value, as per the semantics of DIP.execute. A DIM engine that does support and allow execution of such an executable resource shall return a true value and the output of the executable resource can be observed.

NOTE 1 While the executable results can be observed, the equivalence of the executable results is not actually part of the DIP conformance, since that is outside the scope of DIP.

NOTE 2 The actual mechanisms for the execution of the executable resource are outside the scope of DIP and DIP conformance testing.

NOTE 3 The source code in standard ANSI C for the executable resource is also provided. This could be used to adapt the test case for other platforms.

15.4.2.6 DIP.getExternalData

The DIP.getExternalData DIBO allows a User to select resources located external to the DI. The result is dependent on the selection made by the User. Thus test cases for DIP.getExternalData will also specify the pre-defined resources (provided as part of the test case data) to be selected in order to produce known results.

In addition, the implementer of `DIP.getExternalData` may allow the User selection of the resource to be done in a manner appropriate to the MPEG-21 environment in which the DIM engine is executing. The conformance testing is testing only the correct selection of the DIP Objects according to the User interactions, not the manner of such selection.

15.4.2.7 DIP.getObjects

The `DIP.getObjects` DIBO allows a User to select one or more DIP Objects of given Object Types. The result is dependent on the selection made by the User. Thus test cases for `DIP.getObjects` will also specify the pre-defined Objects to be selected in order to produce known results.

In addition, the implementer of `DIP.getObjects` may allow the User selection of the resource to be done in a manner appropriate to the MPEG-21 environment in which the DIM engine is executing. The conformance testing is testing only the correct entering of values according to the User interactions, not the manner of such data entry.

15.4.2.8 DIP.getValues

The `DIP.getValues` DIBO allows a User to enter one or more values of given data types. The result is dependent on the values entered by the User. Thus test cases for `DIP.getValues` will also specify the pre-defined data values to be entered in order to produce known results.

In addition, the implementer of `DIP.getValues` may allow the User entry of values to be done in a manner appropriate to the MPEG-21 environment in which the DIM engine is executing. The conformance testing is testing only the correct selection of the resource according to the User interactions, not the manner of such selection.

15.4.2.9 DIP.play

The `DIP.play` semantics are to render into a transient and directly perceivable representation a given Component or Descriptor. The manner of playing the Component or Descriptor may be implemented as appropriate to the MPEG-21 environment in which the DIM engine is executing. `DIP` itself does not specify any requirements in regards to support for associated resource types or how to render associated resource types. From this perspective the functionality of the `DIP.play` DIBO is to provide the normative interface by which a DIM author can request a Component or Descriptor to be played (that might or might not be supported) by the underlying MPEG-21 environment.

For `DIP.play` test cases for a finite set of resource types are provided.

Confirming that the Component or Descriptor has been rendered into a transient and directly perceivable representation will be somewhat subjective on the part of the tester and dependent on the particular specifications of the DIM engine implementation.

15.4.2.10 DIP.print

The `DIP.print` semantics are to render into a fixed and directly perceivable representation a Component or Descriptor. `DIP.print` is thus similar to `DIP.play` (15.4.2.9) except that the representation is fixed rather than transient.

15.4.3 Testing DIM engine conformance

Such processing, that is DIM execution, can be tested by appropriately authored DIDL documents and DIMs and checking the results of a DIM engine against the expected results.

The reference DIP engine is linked to a DIM engine as part of its processing of a DIDL document containing DIP. Hence the reference DIP engine in conjunction with the test cases specified in L.2 can be used to test a DIM engine for DIM engine conformance. The general procedure for testing DIM engine conformance is thus the same as that for testing DIP engine conformance (see Figure 5 — Testing DIP engine conformance).

NOTE Conformance for processing of standard ECMAScript language elements and the DOM APIs is defined by those specifications.

15.5 J-DIXO conformance

15.5.1 Introduction

The following subclauses outline how to check conformance for MPEG-21 DIDs that include J-DIXOs as well as compliance of MPEG-21 implementations that support J-DIXOs.

15.5.2 J-DIXO inclusion in DID Conformance

DIDs that include J-DIXOs shall follow all the rules outlined in subclause C.3 of ISO/IEC 21000-10:2006. This conformance is covered in 15.2.

15.5.3 J-DIXO conformance

There are two parts to J-DIXO conformance. The conformance of the terminal that supports J-DIXO execution and the conformance of what constitutes J-DIXO data.

15.5.3.1 Terminal Conformance

15.5.3.1.1 J-DIXO Invocation

The invocation rules of J-DIXOs are covered in subclause C.2 of ISO/IEC 21000-10:2006. All implementations that support J-DIXOs shall adhere to these rules to be conformant.

15.5.3.1.1.1 Testing

J-DIXO processing functionality of implementations are tested using appropriately authored reference DIDs that include J-DIXOs. This is a necessary but not a sufficient condition. All the rules that are outlined in subclauses C.2 and C.3 of ISO/IEC 21000-10:2006 shall be adhered to.

15.5.3.1.2 Java Platform Compliance

The Java Platform compliance is out of scope of this document. The MPEG-J based model for calling J-DIXOs is in Annex D of ISO/IEC 21000-10:2006.

15.5.3.1.3 J-DIBO Implementation

For full conformance all the J-DIBOs (Java data type bindings for DIML object types, Java interface bindings for DIBOs, and J-DIBO factory) as specified in Annex B (Java bindings for Digital Item Operations) of ISO/IEC 21000-10:2006 shall be implemented. The behaviour, return values and the methods signatures as specified in Annex B of ISO/IEC 21000-10:2006 shall be adhered to. For conformance to a DIP profile all the J-DIBOs as specified in the profile shall be implemented.

15.5.3.1.3.1 Testing

J-DIBO implementations can be tested using appropriately authored DIDs that include J-DIXOs which use various J-DIBOs. This is a necessary but not a sufficient condition.

15.5.3.2 J-DIXO language Conformance

A J-DIXO Java program (that implements the J-DIXO Java interface) or a J-DIXO helper class shall be compliant with the Java language specification that is supported by the Java Platform for which it is intended. Additionally, J-DIXOs can call the Java APIs (J-DIBOs) specified in Annex B of ISO/IEC 21000-10:2006 (org.iso.mpeg.mpeg-21.mpegj package).

15.5.3.2.1.1 Testing

J-DIXOs can be run through reference DIP implementation to test compliance. This is a necessary but not a sufficient condition.

16 DIP Amd.1 Conformance

ISO/IEC 21000-10:2006/Amd.1:2006 Conformance is two-fold: conformance of the terminal that support C++ bindings and conformance of what constitutes C++ executable data that makes use of C++ bindings.

16.1 Terminal conformance: C++ binding implementation

The C++ data type bindings for DIML object types, the C++ DIBO factory interface, the C++ global environment interface and the C++ interface bindings for DIBOs, as specified in ISO/IEC 21000-10:2006/Amd.1:2006, shall be implemented (i.e., the behavior, the return values and the methods signatures shall be adhered to).

16.2 C++ executable conformance

A C++ executable (that interacts with the DIP environment) shall be compliant with the C++ language specification that is supported by the C++ platform for which it is intended. Additionally, such C++ executables shall be able to use data type bindings for DIML object types and call C++ bindings for DIBOs as specified in ISO/IEC 21000-10:2006/Amd.1:2006.

In Annex M, test DIDs and C++ functions are defined for the following identified DIP Amd.1 conformance points:

- C++ data type bindings for DIML object types.
- C++ DIBO factory interface.
- C++ global environment interface.
- C++ interface bindings for DIBOs.

17 Event Reporting Conformance

17.1 Event Report Request Description Conformance

An Event Report Request description is conformant only if it is well-formed XML and meets all the description constraints specified in clause 7 of ISO/IEC 21000-15:2006. These constraints are specified in text and schemas in various clauses and subclauses throughout ISO/IEC 21000-15:2006.

17.2 Event Report Description Conformance

An Event Report description is conformant only if it is well-formed XML and meets all the description constraints specified in clause 8 of ISO/IEC 21000-15:2006. These constraints are specified in text and schemas in various clauses and subclauses throughout ISO/IEC 21000-15:2006.

17.3 Event Report Request Parser Conformance

An Event Report Request Parser is defined as a module that takes a file as input and produces as output either 1 (if the input file is a conformant Event Report Request Description) or 0 (if the input file is not a conformant Event Report Request Description).

An Event Report Request Parser is conformant only if, given the inputs given in Annex N it produces the corresponding outputs from Annex N.

17.4 Event Report Parser Conformance

An Event Report Parser is defined as a module that takes a file as input and produces as output either 1 (if the input file is a conformant Event Report Request Description) or 0 (if the input file is not a conformant Event Report Description).

An Event Report Parser is conformant only if, given the inputs given in Annex N it produces the corresponding outputs from Annex N.

17.5 Draft List of Conformance Tests

Embedding an ER-R within a Digital Item

Embedding an ER within a Digital Item

An ER-R with some mandatory elements missing:

- no recipient
- no conditions
- no reportable data items
- etc..

An ER-R that is compliant with the schema and has the absolute bare minimum of elements.

An ER-R that is compliant with the schema and contains all possible elements – this should include:

- 1) An example of each condition type
- 2) An embedded ER-R in the ER Specification
- 3) Custom reportable data
- 4) DI Metadata reportable data
- 5) Selecting a DI
 - i) From Metadata elements
 - ii) Using XPATH
 - iii) using DII

Given an ER-R, give the resulting ER (need to indicate that the conditions stated are all fulfilled before the ER is created).

18 Binary Format Conformance

MPEG-21 Binary Format ISO/IEC 21000-16:2005 is based on the generic BiM Format (MPEG-B Part 1), whose conformance can be found in [ISO/IEC 23001-1 clause 9].

The list of reference bitstreams for MPEG-21 Binary Format is given in Annex N.

19 Fragment Identification Conformance

19.1 Introduction

The following subclauses describe conformance points and test suites for ISO/IEC 21000-17:2006.

19.2 Syntax Conformance

A URI fragment identifier is conformant only if:

- a) it is conformant to IETF RFC 3986
- b) All syntactic rules that are normatively defined in ISO/IEC 21000-17:2006 are met.

19.3 FID Parser conformance

An FID parser is defined as a module that takes the following inputs:

- one or more Logical Models or an Internet Media Type
- a fragment identifier,

and produces as output either 1 (if the input fragment identifier is syntactically conformant to ISO/IEC 21000-17:2006) or 0 (if the input fragment identifier is not syntactically conformant to ISO/IEC 21000-17:2006).

An FID Parser is not conformant if for any of the input specified in the tests provided in Annex P, it does not produce the corresponding outputs from Annex P.

19.4 Logical Model conformance

A Logical Model Schema is conformant only if it is a well-formed XML document that validates against the schema for Logical Model representation specified in ISO/IEC 21000-17:2006.

19.5 Logical Model Parser conformance

A Logical Model Schema parser is defined as a module that takes a Logical Model Schema as an input and produces as output either 1 (if the input Logical Model Schema is conformant to ISO/IEC 21000-17:2006) or 0 (if the input Logical Model Schema is not a conformant to ISO/IEC 21000-17:2006).

A Logical Model Schema Parser is not conformant if for any of the inputs given in Annex P, it does not produce the corresponding outputs from Annex P.

Annex A (normative)

Tests for DIDL Documents

A.1 Introduction

See attached 21000-2 DID zip file.

A.2 Test 1

A.2.1 XML declaration

```
<DIDL xmlns="urn:mpeg:mpeg21:2002:02-DIDL-NS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:mpeg21:2003:01-DIA-NS http://127.0.0.1/21000-07-UE.xsd">
  <Declarations>
    <Item id="track04">
      <Condition except="choice_01_select_id_01"/>
      <Component>
        <Resource ref="track02.mp3" mimeType="audio/mp3"/>
      </Component>
    </Item>
  </Declarations>
  <Descriptor>
    <Statement mimeType="text/plain">My music album</Statement>
  </Descriptor>
  <Component>
    <Resource mimeType="text/xml">FunkyTest <dia:DIA
      xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <dia:Description xsi:type="dia:UsageEnvironmentType">
        <dia:UsageEnvironment xsi:type="dia:TerminalCapabilitiesType">
          <dia:TerminalCapabilities
            xsi:type="dia:InputOutputCapabilitiesType">
            <dia:Display>
              <dia:Resolution horizontal="176" vertical="144"/>
            </dia:Display>
          </dia:TerminalCapabilities>
        </dia:UsageEnvironment>
        <dia:UsageEnvironment xsi:type="dia:NetworkCharacteristicsType">
          <dia:NetworkCharacteristics
            xsi:type="dia:NetworkCapabilityType" maxCapacity="64000"/>
        </dia:UsageEnvironment>
      </dia:Description>
      <dia:Description xsi:type="dia:UsageEnvironmentType">
        <dia:UsageEnvironment xsi:type="dia:NetworkCharacteristicsType">
          <dia:NetworkCharacteristics
            xsi:type="dia:NetworkCapabilityType"
              maxCapacity="384000" minGuaranteed="32000"/>
          <dia:NetworkCharacteristics
            xsi:type="dia:NetworkConditionType">
            <dia:AvailableBandwidth maximum="256000" average="80000"
              interval="330"/>
            <dia:Delay packetTwoWay="330" delayVariation="66"/>
            <dia:Error packetLossRate="0.05"/>
          </dia:NetworkCharacteristics>
        </dia:UsageEnvironment>
      </dia:Description>
    </Resource>
  </Component>
</DIDL>
```

```

        </dia:DIA>
    </Resource>
</Component>
<Anchor>
    <Fragment>test</Fragment>
</Anchor>
</Declarations>
<Container id="test">
    <Item id="myitem">
        <Descriptor>
            <Statement mimeType="text/xml">
                <XML>before XML2<XML2>My music album</XML2>after XML2</XML>
            </Statement>
        </Descriptor>
        <Choice choice_id="choice_01" minSelections="1" maxSelections="3">
            <Selection select_id="choice_01_select_id_01"/>
        </Choice>
        <Item id="track01">
            <Condition require="choice_01_select_id_01"/>
            <Component id="mycomp">
                <Resource ref="track01.mp3" mimeType="audio/mp3"/>
            </Component>
        </Item>
        <Item id="track02">
            <Condition except="choice_01_select_id_01"/>
            <Component>
                <Resource ref="track02.mp3" mimeType="audio/mp3"/>
            </Component>
        </Item>
        <Item id="track03">
            <Component>
                <Resource ref="track03.mp3" mimeType="audio/mp3"/>
                <Anchor>
                    <Fragment>test</Fragment>
                </Anchor>
            </Component>
        </Item>
        <Annotation target="#track02">
            <Descriptor>
                <Statement mimeType="text/plain">My music album</Statement>
            </Descriptor>
        </Annotation>
        <Annotation target="#mycomp">
            <Anchor>
                <Fragment>test</Fragment>
            </Anchor>
        </Annotation>
        <Annotation target="#myitem">
            <Assertion target="#choice_01" false="choice_01_select_id_01"/>
        </Annotation>
    </Item>
</Container>
</DIDL>

```

A.2.2 Validation result

Output: 0.

Error occurred in the line 94 of the XML declaration.

Validation rule number R-DID-xxxxx

Explanation: <Assertion>: The total number of select_id values defined in the associated Choice, minus the number of select_id values listed in the false attribute, must be greater than or equal to the minSelections attribute value in the associated Choice.

Output from the reference software:

```
org.iso.mpeg.mpeg21.did.validation.DIDValidationException:
***** Validation Error *****
Validation rule:
<Assertion>: The total number of select_id values defined in the associated Choice,
minus the number of select_id values listed in the false attribute, must be greater than
or equal to the minSelections attribute value in the associated Choice.

<?xml version='1.0' ?>
  <didl:Assertion target="#choice_01" except="choice_01_select_id_01"
xmlns:didl="urn:mpeg:mpeg21:2002:02-DIDL-NS" />
*****
```

A.3 Test 2

A.3.1 XML declaration

```
<?xml version="1.0" encoding="UTF-8"?>
<DIDL xmlns="urn:mpeg:mpeg21:2002:02-DIDL-NS">
  <Declarations>
    <Item id="track_01">
      <Component>
        <Resource ref="track02.mp3" mimeType="audio/mp3"/>
      </Component>
    </Item>
  </Declarations>
  <Item>
    <Item>
      <Reference target="#track_011"/>
    </Item>
  </Item>
</DIDL>
```

A.3.2 Validation result

Output: 0.

Error occurred in the line 12 of the XML declaration.

Validation rule number R-DID-xxxxx

Explanation: <Reference>: The target attribute must evaluate to a single XML node. <Reference>: The target attribute must evaluate to DIDL element.

Output from the reference software:

```
org.iso.mpeg.mpeg21.did.validation.DIDValidationException:
***** Validation Error *****
Validation rule:
<Reference>: The target attribute must evaluate to a single XML node.
<Reference>: The target attribute must evaluate to DIDL element.

<?xml version='1.0' ?>
  <didl:Reference target="#track_011" xmlns:didl="urn:mpeg:mpeg21:2002:02-DIDL-NS" />
*****
```

A.4 Test 3

A.4.1 XML declaration

```
<?xml version="1.0" encoding="UTF-8"?>
<DIDL xmlns="urn:mpeg:mpeg21:2002:02-DIDL-NS">
  <Declarations>
    <Descriptor id="track_01">
      <Component>
        <Resource ref="track02.mp3" mimeType="audio/mp3"/>
      </Component>
    </Descriptor>
  </Declarations>
  <Item>
    <Item>
      <Reference target="#track_01"/>
    </Item>
  </Item>
</DIDL>
```

A.4.2 Validation result

Output: 0.

Error occurred in the line 12 of the XML declaration.

Validation rule number R-DID-xxxxx

Explanation: <Reference>: The fully qualified name of the referent element must match the fully qualified name of the referring element.

Output from the reference software:

```
org.iso.mpeg.mpeg21.did.validation.DIDValidationException:
***** Validation Error *****
Validation rule:
<Reference>: The fully qualified name of the referent element must match the fully
qualified name of the referring element.

<?xml version='1.0' ?>
  <didl:Reference target="#track_01" xmlns:didl="urn:mpeg:mpeg21:2002:02-DIDL-NS" />
*****
```

A.5 Test 4

A.5.1 XML declaration

```
<?xml version="1.0" encoding="UTF-8"?>
<DIDL xmlns="urn:mpeg:mpeg21:2002:02-DIDL-NS" xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS"
xmlns:dip="urn:mpeg:mpeg21:2003:01-DIP-NS">
  <Declarations>
    <Item id="track_01">
      <Descriptor id="descriptor_01">
        <Statement mimeType="text/xml">
          <dii:Identifier>urn:grid:a1-abcde-9873216540-f</dii:Identifier>
        </Statement>
      </Descriptor>
      <Descriptor id="descriptor_02">
        <Statement mimeType="text/xml">
          <dii:Identifier>urn:grid:a1-abcde-9873216540-f</dii:Identifier>
        </Statement>
      </Descriptor>
    </Item>
  </Declarations>
</DIDL>
```