



**INTERNATIONAL STANDARD ISO/IEC 15938-1:2002/Amd.1:2005**  
**TECHNICAL CORRIGENDUM 1**

Published 2005-11-15

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION  
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

## **Information technology — Multimedia content description interface —**

### **Part 1: Systems**

AMENDMENT 1: Systems extensions

TECHNICAL CORRIGENDUM 1

*Technologies de l'information — Interface de description du contenu multimédia —*

*Partie 1: Systèmes*

*AMENDEMENT 1: Extensions des systèmes*

*RECTIFICATIF TECHNIQUE 1*

Technical Corrigendum 1 to ISO/IEC 15938-1:2002/Amd.1:2005 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

In subclause 7.1, replace:

The schema component codes (type codes, element codes or attribute codes) are accessible through all these schemas. However codes are constructed differently depending on which schema they are defined. The initial schema aggregates all schema components possibly coming from different namespaces in a single code space. On the contrary, additional schemas contain only schema components which are defined in their namespace.

with:

The schema component codes (type codes, element codes or attribute codes) are accessible through all these schemas. However codes are constructed differently depending on which schema they are defined in. The initial schema aggregates all schema components possibly coming from different namespaces in a single code space. On the contrary, additional schemas contain only schema components which are defined in their namespace.

In subclause 7.2.1, replace:

Transmission of additional schema is specified for two different use cases: The retrieval of schema information in binary format from a location indicated by a URI, the transmission of schema information in a binary description stream jointly or not with the transmission of a description. In the latter case there is a requirement that all schema information needed for the decoding of a fragment of the transmitted description must have been received before such fragment arrives.

with:

Transmission of additional schemas is specified for two different use cases: The retrieval of schema information in binary format from a location indicated by a URI, the transmission of schema information in a binary description stream jointly or not with the transmission of a description. In the latter case there is a requirement that all schema information needed for the decoding of a fragment of the transmitted description must have been received before such fragment arrives.

Replace the following row in the table of subclause 7.2.2:

<b>ReservedBitsZero</b>	FeatureFlags_ Length*8-6	bslbf
-------------------------	--------------------------	-------

with:

<b>ReservedBitsZero</b>	FeatureFlags_Le ngth*8-7	bslbf
-------------------------	--------------------------	-------

Revise the table in subclause 7.2.4 as highlighted below:

AdditionalSchemaConfig () {	Number of bits	Mnemonic
<b>NumberOfAdditionalSchemas</b>	8+	vluimsbf8
<b>NumberOfKnownAdditionalSchemas</b>	8+	vluimsbf8
for (int t=1;t<NumberOfKnownAdditionalSchemas-1;t++){		
<b>KnownAdditionalSchemaID</b>	8+	
<b>AdditionalSchemaURI_Length[KnownAdditionalSchemaID]</b>	8+	vluimsbf8
<b>AdditionalSchemaURI[KnownAdditionalSchemaID]</b>	8* AdditionalSchema URI_Length [KnownAdditional SchemaID]	bslbf

<b>BinaryLocationHint_Length[KnownAdditionalSchemaID]</b>	8+	vluimsbf8
<b>BinaryLocationHint[KnownAdditionalSchemaID]</b>	8*BinaryLocationHint_Length[KnownAdditionalSchemaID]	bslbf
<b>NumberOfTypeCodecs[KnownAdditionalSchemaID]</b> for (i=0; i< NumberOfTypeCodecs[KnownAdditionalSchemaID]; i++) {	8+	vluimsbf8
<b>TypeCodecURI_Length[KnownAdditionalSchemaID][i]</b>	8+	vluimsbf8
<b>TypeCodecURI[KnownAdditionalSchemaID][i]</b>	8*TypeCodecURI_Length[KnownAdditionalSchemaID][i]	bslbf
<b>NumberOfTypes[KnownAdditionalSchemaID][i]</b> for (j=0; j< NumberOfTypes[KnownAdditionalSchemaID][i]; j++) {	8+	vluimsbf8
<b>TypeIdentificationCode[KnownAdditionalSchemaID][i][j]</b>	8+	vluimsbf8
}		
}		
ExternallyCastableTypeTable(KnownAdditionalSchemaID)		
ExternallySubstitutableElementTable(KnownAdditionalSchemaID)		
}		
<b>SchemaEncodingMethod</b>	8	blsbf
ExternallyCastableTypeTable(InitialSchema)		
ExternallySubstitutableElementTable(InitialSchema)		
<b>ReservedBitsZero</b>	7	blsbf
}		

Replace the following row in the table in subclause 7.2.5:

NumberOfAdditionalSchemas	Indicates the number of schemas that can be transmitted and that are not declared in the list of schemaURI. If additional schemas are not supported, this value is set to zero.
---------------------------	---

with:

NumberOfAdditionalSchemas	Indicates the number of schemas that can be transmitted and that are not declared in the list of schemaURI. If additional schemas are not supported, this value is set to zero. If additional schemas are supported then the value of GlobalSchemaID = NumberOfSchemas is reserved for a virtual namespace for attributes that do not belong to any namespace. The value of GlobalSchemaID = NumberOfSchemas + NumberOfAdditionalSchemas - 1 is reserved for ISO usage.
---------------------------	--

Replace the following row in the table in subclause 7.2.5:

KnownAdditionalSchemaID	Identifies a schema known to be updated in the bistream. This identifier shall only address an additional schema i.e. its value shall be superior to 'NumberOfSchemas-1'
-------------------------	--

with:

KnownAdditionalSchemaID	Identifies a schema known to be updated in the bitstream. This identifier shall only address an additional schema. The value KnownAdditionalSchemaID = NumberOfSchemas shall be reserved for attributes that do not belong to any namespace. The value of KnownAdditionalSchemaID = NumberOfSchemas + NumberOfAdditionalSchemas - 1 shall be reserved for ISO Use.
-------------------------	--

In subclause 7.7.1, replace:

A schema update unit is composed of a namespace identifier, a set of code tables to represent global elements, global types and global attributes, followed by a binary encoded schema carrying the schema components definitions. This binary encoded schema is encoded using a specific profile of BiM specified in subclause 7.7.8 using a simple XML schema for schema encoding has been defined for this purpose in this specification.

with:

A schema update unit is composed of a namespace identifier, a set of code tables to represent global elements, global types and global attributes, followed by a binary encoded schema carrying the schema components definitions. This binary encoded schema is encoded using a specific profile of BiM specified in subclause 7.7.8. For that purpose a simple XML schema for schema encoding has been defined in this specification.

Replace subclause 8.5.2.4.5 with the following text:

**8.5.2.4.5 Wildcard transition behaviour**

**8.5.2.4.5.1 Introduction**

Wildcard transitions, when crossed, specify to the decoder that an element not known a priori is present in the description. The decoder shall execute the following decoding procedure or skip its decoding if the element in the schema is unknown.

**8.5.2.4.5.2 AnyElementDecoding Syntax**

AnyElementDecoding() {	Number of bits	Mnemonic
<b>GlobalElementSchemaID</b>	ceil( log2( NumberOfSchemas + NumberOfAdditionalSchemas ) )	uimsbf
<b>AnyElement_Length</b>	5+	vluimsbf
<b>Any_SBC_Operand_Selector</b>	5+	vluimsbf
If (inPayloadDecoding()) {		
Element(ChildrenSchemaMode, theAnyType)		
}		
}		