



**INTERNATIONAL STANDARD ISO/IEC 15938-12:2008**  
**TECHNICAL CORRIGENDUM 2**

Published 2010-02-15

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 12: Query format**

#### **TECHNICAL CORRIGENDUM 2**

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

*Partie 12: Format de requête*

*RECTIFICATIF TECHNIQUE 2*

Technical Corrigendum 2 to ISO/IEC 15938-12 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Replace:

The `QueryCondition` element, of the `QueryConditionType` type, is the part of the Input Query Format where the user specifies the properties of the media or the metadata to be retrieved. The `QueryCondition` element is defined as a sequence of desired target media types, an `EvaluationPath` element and a condition tree, which can be a simple (`Condition` element) or a Join operation (`Join` element). The sequence of the target media types declares the `ggdesired` MIME types which the user expects as a result. The `EvaluationPath` element (optional) declares an XPath expression, which specifies the node of the metadata fragment related to the evaluation item being addressed (`//VideoSegment` for instance). It also determines the structure of the output; one result item will be returned for each evaluation item if it matches the condition. If the `EvaluationPath` element is not specified, the output result shall be provided as a collection of multimedia contents, as stored in the repository each of which satisfies the query condition. The condition tree can be specified using the `Condition` element, of the `BooleanExpressionType` type, which is the parent abstract type of all the types resulting in a value in the range of [0..1]. A more complex condition tree can be specified using the `Join` element. The `Join` operation allows the definition of filtering conditions which act over two sets of multimedia objects.

with the following (where changes to existing text are highlighted in grey):

The `QueryCondition` element, of the `QueryConditionType` type, is the part of the Input Query Format where the user specifies the properties of the media or the metadata to be retrieved. The `QueryCondition` element is defined as a sequence of desired target media types, an `EvaluationPath` element and a condition tree, which can be a simple (`Condition` element) or a Join operation (`Join` element). The sequence of the target media types declares the `ggdesired` MIME types which the user expects as a result. The `EvaluationPath` element (optional) declares an XPath expression, which specifies the node of the metadata fragment related to the evaluation item being addressed (`//VideoSegment` for instance). It also determines the structure of the output; one result item will be returned for each evaluation item if it matches the condition. If the `EvaluationPath` element is not specified, the output result shall be provided as a collection of multimedia contents, as stored in the repository each of which satisfies the query condition.

If the `QueryCondition` element does not appear within the Input element, or if the `QueryCondition` element does not contain neither a `Condition` nor a `Join` element, it is considered to be an Empty Query asking to retrieve all records in the database.

The condition tree can be specified using the `Condition` element, of the `BooleanExpressionType` type, which is the parent abstract type of all the types resulting in a value in the range of [0..1]. A more complex condition tree can be specified using the `Join` element. The `Join` operation allows the definition of filtering conditions which act over two sets of multimedia objects.

Replace:

```
<complexType name="QueryConditionType">
  <sequence>
    <element name="EvaluationPath" type="mpqf:xPathType" minOccurs="0"/>
    <element name="TargetMediaType" type="mpqf:mimeType" minOccurs="0"
maxOccurs="unbounded"/>
    <choice>
      <element name="Join" type="mpqf:JoinType"/>
      <element name="Condition" type="mpqf:BooleanExpressionType"/>
    </choice>
  </sequence>
</complexType>
```

with the following (where changes to existing text are highlighted in light grey):

```
<complexType name="QueryConditionType">
  <sequence>
    <element name="EvaluationPath" type="mpqf:xPathType" minOccurs="0"/>
    <element name="TargetMediaType" type="mpqf:mimeType" minOccurs="0"
maxOccurs="unbounded"/>
    <choice minOccurs="0">
      <element name="Join" type="mpqf:JoinType"/>
      <element name="Condition" type="mpqf:BooleanExpressionType"/>
    </choice>
  </sequence>
</complexType>
```

Pages 93 and 94, B.2

Replace:

```
<Term href="100.3.6">
  <Name>QueryType</Name>
  <Term href="100.3.6.1">
    <Name>QueryByMedia</Name>
  </Term>
  <Term href="100.3.6.2">
    <Name>QueryByDescription</Name>
  </Term>
  <Term href="100.3.6.3">
    <Name>QueryByFreeText</Name>
  </Term>
  <Term href="100.3.6.4">
    <Name>QueryByXQuery</Name>
  </Term>
  <Term href="100.3.6.5">
    <Name>QueryByRelevanceFeedback</Name>
  </Term>
  <Term href="100.3.6.6">
    <Name>QueryByFeatureRange</Name>
  </Term>
  <Term href="100.3.6.7">
    <Name>SpatialQuery</Name>
  </Term>
  <Term href="100.3.6.8">
    <Name>TemporalQuery </Name>
  </Term>
</Term>
```