INTERNATIONAL STANDARD

ISO/IEC 24800-2

First edition
2011-07-01

Information technology — JPSearch —

Part 2: Registration, identification and management of schema and ontology

Technologies de l'information — JPSearch —

Partie 2: Enregistrement, identification et gestion des schémas et de l'ontologie

Reference number
ISO/IEC 24800-2:2011(E)

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

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 24800-2 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 24800 consists of the following parts, under the general title *Information technology — JPSearch*:

— *Part 1: System framework and components*

— *Part 2: Registration, identification and management of schema and ontology*

— *Part 3: Query format*

— *Part 4: File format for metadata embedded in image data (JPEG and JPEG 2000)*

— *Part 5: Data interchange format between image repositories*

— *Part 6: Reference software*

Introduction

This part of ISO/IEC 24800 provides a standardized set of technologies for metadata representation, querying and management of images. It specifies the JPSearch's Core Metadata Schema as the cornerstone of metadata interoperability in ISO/IEC 24800. It also specifies the structure and rules to which any metadata annotation of images must conform in order to be considered valid within a JPSearch compliant system.

In addition to the definition of JPSearch Core Metadata Schema, ISO/IEC 24800 provides a mechanism which allows a JPSearch compliant system taking profit from proprietary or community-specific metadata schemas. A translation rules language is defined, allowing the publication of machine-readable translations between metadata terms belonging to proprietary metadata schemas and metadata terms in the JPSearch Core Metadata Schema. Users can choose which metadata language to use in a JPSearch-based interaction (annotation, querying, etc.) if the proper translations are available.

In order to specify the issues in a detailed manner in this document, this part of ISO/IEC 24800 first provides the fundamental information including scope definition, description of terms and definitions, and conventions that are necessary to understand this document. The definition of JPSearch Core Metadata Schema is described in the context of XML structure. Management of information regarding other metadata schema is also described in respect of registration, maintenance, and translation rules.

Information technology — JPSearch —

Part 2:
Registration, identification and management of schema and ontology

1 Scope

This part of ISO/IEC 24800 specifies a series of interfaces to allow disparate systems an interoperable management of image repositories. It also specifies the general rules which govern the usage of metadata in JPSearch and provides a specification which

— provides rules for the representation of image metadata descriptions, consisting of the definition of the JPSearch Core Metadata Schema,

— provides rules for the publication of machine-readable translations between metadata terms belonging to proprietary metadata schemas and metadata terms in the JPSearch Core Metadata Schema, and

— provides rules for the registration and request of metadata schemas and its translation rules or links to them.

JPSearch is an extensible standard. The normative method of extending the structures and rules beyond the JPSearch Core Metadata Schema is provided in this part of ISO/IEC 24800.

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.

XML, Extensible Markup Language (XML) 1.0 (Fifth Edition), 26 November 2008, available at http://www.w3.org/TR/xml/

XML Schema Part 1: Structures Second Edition, W3C Recommendation, 28 October 2004, available at http://www.w3.org/TR/xmlschema-1/

XML Schema Part 2: Datatypes Second Edition, W3C Recommendation, 28 October 2004, available at http://www.w3.org/TR/xmlschema-2/

XPath, XML Path Language, W3C Recommendation, 16 November 1999, available at http://www.w3.org/TR/xpath

NOTE These documents are maintained by the W3C (http://www.w3.org).

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
Core schema
metadata basis supporting interoperability during search among multiple image retrieval systems

NOTE The core schema is used by clients to formulate in combination with the JPEG Query Format search requests to JPSearch compliant search systems. Note, that only metadata described by the core schema is guaranteed to be processed by JPSearch compliant systems.

3.2
Translation rules
machine-readable declaration of the semantic and syntactic mappings between a proprietary metadata schema and the JPSearch's core metadata schema

3.3
JPSearch registration authority
Organization that supports the registration and request of metadata schemas and its translation rules or links to them

NOTE It is necessary that every participating content provider registers their schema and translation rules or a link to them at this authority. In case the JPSearch compliant retrieval system is operated in offline mode, the necessary information (target schema, translation rules, etc.) should be available at the respective system itself.

4 Conventions

4.1 Naming convention

In order to specify the JPSearch Core metadata description scheme, this Part of ISO/IEC 24800 uses constructs provided by XML such as "element" and "complexType." The names associated to these constructs are created on the basis of the following conventions:

If the name is composed of multiple words, the first letter of each word is capitalized, with the exception that the capitalization of the first word depends on the type of construct and is described below.

- Element naming: the first letter of the first word is capitalized (e.g. `Identifier` element of `JPSearchCoreType`).
- Attribute naming: the first letter of the first word is not capitalized (e.g. jpsearchID attribute of `ManagementType` type).
- complexType naming: the first letter of the first word is capitalized, and the suffix "Type" is used at the end of the name (e.g. `JPSearchCoreType`).
- simpleType naming: the first letter of the first word is not capitalized, the suffix "Type" may be used at the end of the name (e.g. `xPathType`).

4.2 Document convention

The syntax of each description is specified using the constructs provided by XML as defined in XML, XML Schema Part 1, and XML Schema Part 2, and is presented in this document using a specific font and background as shown in the following example:

```
<complexType name="ExampleType">
    <sequence>
       <element name="Element1" type="string"/>
    </sequence>
    <attribute name="attribute1" type="string" default="attrvalue1"/>
</complexType>
```

The semantics of each description tool is specified in text using a table format, where each row contains the name and a definition of a type, element or attribute as shown in the following example:

Name	Definition
ExampleType	Specifies an …
element1	Describes the …
attribute1	Describes the …

4.3 Wrapper of the schema

The description examples and syntax of description tools specified in this document assume that a schema wrapper is provided which identifies the XML Schema namespace (XML Schema) and JPSearch namespace:

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:JPCore="JPSearch:schema:coremetadata"
        targetNamespace="JPSearch:schema:coremetadata"
        elementFormDefault="qualified"
        attributeFormDefault="unqualified">
```

The following tag is used to close the schema:

```
</schema>
```


5 JPSearch Core Metadata Schema

5.1 General

This clause targets on the definition of the complex types for the registration process of the schema, its translation rules and contact information. The process of registration is mandatory for all schemas that can be addressed within a retrieval operation. The standard supports two scenarios. First, a global authority for schemas and their translation rules will be established where all JPSearch compliant retrieval applications can obtain the necessary information. The list of ISO registration authorities is listed at http://www.iso.org/iso/maintenance_agencies. Second, in case the retrieval application operates in offline mode, the schema and their translation rules shall be located at the application itself.

JPSearch Core Metadata Schema contains four types: `PersonNameType`, `SourceType`, `PublisherType` and `JPSearchCoreType`. Moreover, in order to support `JPSearchCoreType`, several types are defined: `RightsDescriptionType`, `PlaceType`, `PersonType`, `OrganizationType`, `EventType`, `ObjectType`, `RegionOfInterestType`, `RegionLocatorType`, `ExternalDescriptionType`, `ControlledRatingTermType`, `ImageIdentifierType` and `GPSPositionType`.

5.2 JPSearchCoreType

5.2.1 Introduction

The `JPSearchCoreType` type is devised in order to describe the information about an image in metadata layer. At the same time, as JPSearch core metadata is utilized for image search among the set of images that are described by using heterogeneous metadata schemes, `JPSearchCoreType` contains the most important fields in JPSearch core metadata.

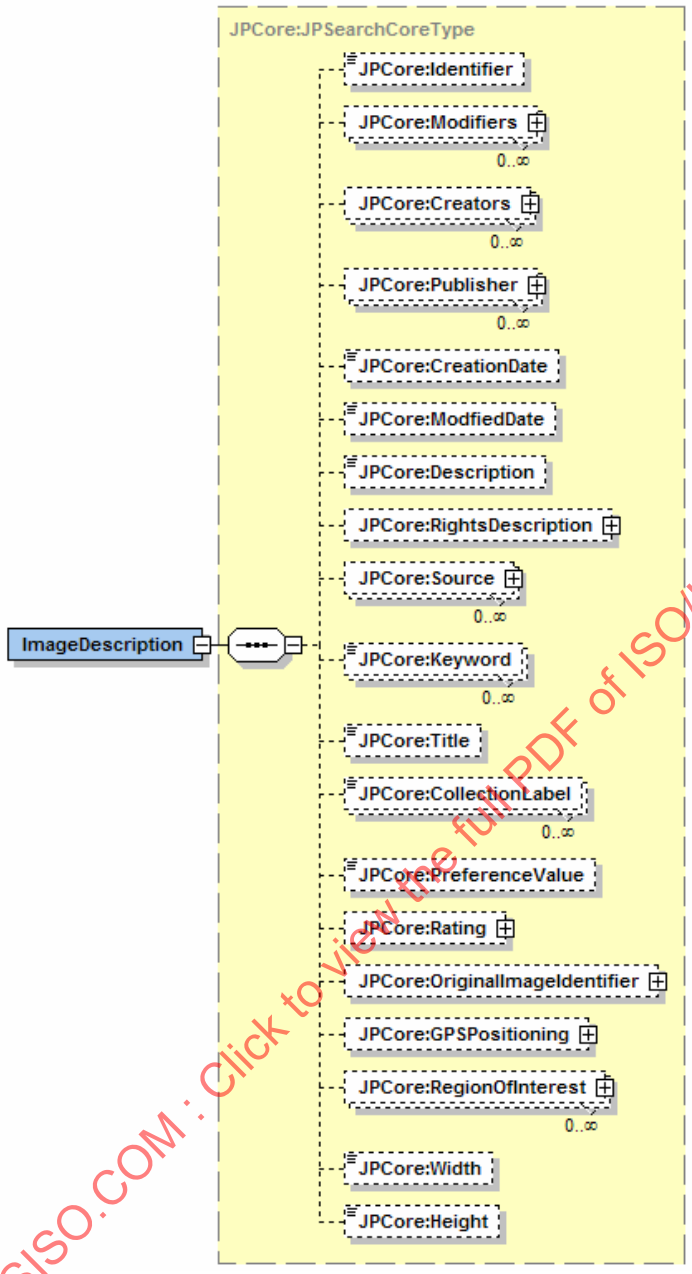

Figure 1 — Diagram representing the JPSearchCoreType

5.2.2 Syntax

```
<element name="ImageDescription" type="JPCore:JPSearchCoreType"/>
<complexType name="JPSearchCoreType">
   <sequence>
     <element name="Identifier" type="anyURI"
              minOccurs="0"/>
     <element name="Modifiers" type="JPCore:PersonNameType"
              minOccurs="0" maxOccurs="unbounded"/>
     <element name="Creators" type="JPCore:PersonNameType"
              minOccurs="0" maxOccurs="unbounded"/>
     <element name="Publisher" type="JPCore:PublisherType"
              minOccurs="0" maxOccurs="unbounded"/>
     <element name="CreationDate" type="dateTime"
```

```
                minOccurs="0"/>
    <element name="ModifiedDate" type="dateTime"
                minOccurs="0"/>
    <element name="Description" type="string"
                minOccurs="0"/>
    <element name="RightsDescription"
type="JPCore:RightsDescriptionType"
                minOccurs="0"/>
    <element name="Source" type="JPCore:SourceType"
                minOccurs="0" maxOccurs="unbounded"/>
    <element name="Keyword" type="string" minOccurs="0"
                maxOccurs="unbounded"/>
    <element name="Title" type="string" minOccurs="0"/>
    <element name="CollectionLabel" type="string" minOccurs="0"
                maxOccurs="unbounded"/>
    <element name="PreferenceValue" type="integer"
                minOccurs="0"/>
    <element name="Rating"
                type="JPCore:ControlledRatingTermType"
                minOccurs="0"/>
    <element name="OriginalImageIdentifier"
type="JPCore:ImageIdentifierType" minOccurs="0"/>
    <element name="GPSPositioning"
                type="JPCore:GPSPositioningType" minOccurs="0"/>
    <element name="RegionOfInterest"
                type="JPCore:RegionOfInterestType" minOccurs="0"
                maxOccurs="unbounded"/>
    <element name="Width" type="int"/>
    <element name="Height" type="int"/>
  </sequence>
</complexType>
```


5.2.3 Semantic

Semantics of the `JPSearchCoreType` type:

Name	Definition
`JPSearchCoreType`	Specifies information for an image.
`Identifier`	Describes an identifier of the image in the form of a URI. The identifier must be unique.
`Modifiers`	Describes a modifier's name or a list of names who changed the original image resulting in the creation of the image (optional).
`Creators`	Describes a person's name or a list of the names who created the image or made contributions in the creation of the image (optional).
`Publisher`	Describes information about the publishing people or organization of the image

CreationDate	Describes the date when the image is created.
ModifiedDate	Describes the date when the image is modified.
Description	Specifies the content of the image in the form of text.
RightsDescription	Describes the right related information by providing information about existing rights description standard, explanation concerning the standard in free text, and rights description in the form of external information or string value.
Source	Describes a source of the image. It can be another image or an object in the form of such as painting, book and so on.
Keyword	Describes a list of keywords that characterize the image (optional).
Title	Describes the title of the image (optional).
CollectionLabel	Describes user provided labels that can be used for the purpose of collection and categorization of images (optional).
PreferenceValue	Describes the value of the preference of the image in the form of integer value.
Rating	Describes the rating results that should be one of the corresponding controlled terms. The definition of the terms is provided by JPSearch.
OriginalImageIdentifier	Describes the identifier of the original image from which the image is created. Moreover, it can be used for the identifiers that can be created by a particular organization or method.
GPSPositioning	Describes the location of the place shown in the image
RegionOfInterest	Describes the information (e.g., content description, keywords, etc.) of a certain region within the image. Note, the whole image itself can also be described by one RegionOfInterest.

5.2.4 Example

This example shows a complete description of an imaginary image. It has a unique ID, some modifier, creator and publisher information. In addition, the creation and modification date have been annotated. Please note, that subparts of this example are reused at the corresponding type definitions and explained in detail there.

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageDescription xmlns="JPSearch:schema:coremetadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="JPSearch:schema:coremetadata jpcore.xsd">
  <Identifier>urn:unique:identifier:1:2:3
    </Identifier>
  <Modifiers>
```

```
    <GivenName>Jonh</GivenName>
    <FamilyName>Smith</FamilyName>
  </Modifiers>
  <Creators>
    <GivenName>Jonh</GivenName>
    <FamilyName>Smith</FamilyName>
  </Creators>
  <Publisher>
    <PersonName>
      <GivenName>Jonh</GivenName>
      <FamilyName>Smith</FamilyName>
    </PersonName>
    <OrganizationInformation>
      <Name>University of Passau</Name>
      <Address>
        <Name>Innstrasse 43, 94034 Passau, Germany
                </Name>
        <Description>
                  Small city at the border to Austria surrounded
                  by three rivers
                </Description>
      </Address>
    </OrganizationInformation>
  </Publisher>
  <CreationDate>2001-12-17T09:30:47.0Z</CreationDate>
  <ModifiedDate>2001-12-17T09:30:47.0Z</ModifiedDate>
  <Description>This is a sample description and this part can contain
                        an arbitrary unstructured text according to the image
    </Description>
  <RightsDescription>
    <RightsDescriptionInformation>
          http://www.rdstandard.org
      </RightsDescriptionInformation>
    <Description>
          Provides a standard for rights description.
</Description>
    <ActualRightsDescriptionReference>
http://www.rdstandard.org/particulaValueRD
</ActualRightsDescriptionReference>
  </RightsDescription>
  <Source>
    <SourceElementType>Oil Painting</SourceElementType>
    <SourceElement>
      <SourceElementTitle>Monna Lisa</SourceElementTitle>
      <SourceElementDescription>Leonardo da Vinci, Louvre, France
</SourceElementDescription>
    </SourceElement>
    <CreationMethod>Photographing</CreationMethod>
    <CreationDescription>CN 450D</CreationDescription>
  </Source>
  <Keyword>Sardinia</Keyword>
  <Keyword>Italy</Keyword>
  <Keyword>50th JPEG meeting</Keyword>
  <Title>Example Instance document of the JPSearch core schema</Title>
  <CollectionLabel>JPEG Meetings</CollectionLabel>
  <PreferenceValue>1</PreferenceValue>
  <Rating>
    <LabelDefinition>urn:ratingtable:quality</LabelDefinition>
    <LabelValue>urn:ratingtable:perfect</LabelValue>
  </Rating>
```

```
  <GPSPositioning latitude="34" longitude="34" altitude="10"/>
  <RegionOfInterest>
    <RegionLocator>
      <Region dim="2"> 0 0 100 100</Region>
    </RegionLocator>
    <Description>A short description about the selected region
        </Description>
    <Keyword>plenary meeting</Keyword>
    <Title>plenary meeting</Title>
    <ContentDescription>
      <Person>
        <Name>
          <GivenName>Jonh</GivenName>
          <FamilyName>Smith</FamilyName>
        </Name>
      </Person>
      <Object>
        <Name>Laptop</Name>
        <Description>running laptop of UoP
                </Description>
        <Label>urn:laptop:id:1:2:3</Label>
      </Object>
      <Place>
        <Name>meeting room xyz</Name>
        <Description>meeting room at the sardinia JPEG meeting
                </Description>
      </Place>
      <Event>
        <Label>urn:writing:event:1:2</Label>
        <Description>writing some document
                </Description>
      </Event>
    </ContentDescription>
    <ExternalDescription>
      <TagName fromNamespace="urn:mpeg:mpeg7:schema:2004"
fromNamespacePrefix="mpeg7">mpeg7:ColorLayout</TagName>
      <StructuredValue fromNamespace="urn:mpeg:mpeg7:schema:2004">
        <mpeg7:Mpeg7 xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004"
xsi:schemaLocation="urn:mpeg:mpeg7:schema:2004 M7v2schema.xsd">
          <mpeg7:DescriptionUnit xsi:type="mpeg7:ColorLayoutType">
            <mpeg7:YDCCoeff>1</mpeg7:YDCCoeff>
            <mpeg7:CbDCCoeff>2</mpeg7:CbDCCoeff>
            <mpeg7:CrDCCoeff>3</mpeg7:CrDCCoeff>
            <mpeg7:YACCoeff2>1 2</mpeg7:YACCoeff2>
            <mpeg7:CbACCoeff2>1 2</mpeg7:CbACCoeff2>
            <mpeg7:CrACCoeff2>1 2</mpeg7:CrACCoeff2>
          </mpeg7:DescriptionUnit>
        </mpeg7:Mpeg7>
      </StructuredValue>
    </ExternalDescription>
    <ExternalDescription>
      <TagName/>
      <LiteralValue/>
    </ExternalDescription>
  </RegionOfInterest>
  <Width>640</Width>
  <Height>480</Height>
</ImageDescription>
```

5.3 PersonNameType

5.3.1 Introduction

The `PersonNameType` type support the description of human beings within images or image regions. The information that can be provided is simplified to `GivenName` and `FamilyName`.

5.3.2 Syntax

```
<complexType name="PersonNameType">
        <sequence>
             <choice maxOccurs="unbounded">
                  <element name="GivenName" type="string" />
                  <element  name="FamilyName"  type="string"  minOccurs="0"
/>
            </choice>
       </sequence>
         <attribute ref="xml:lang" use="optional" />
</complexType>
```

5.3.3 Semantic

Semantics of the `PersonNameType` type:

Name	Definition
`PersonNameType`	Specifies a list of names that consist of given names or family names. The constituent names for a person's name should be described by using the same language.
`GivenName`	Describes a given name. A number of given names can be defined, if necessary.
`FamilyName`	Describes a family name(optional). A number of family names can be defined, if necessary.
`xml:lang`	Describes the language used for defined names(optional).

5.3.4 Example

Instances of the `PersonNameType` are used on several places within the core schema. The following example shows its use by the `Modifiers` tag. In general the `PersonNameType` allows the description of family- and given names of human beings.

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageDescription xmlns="JPSearch:schema:coremetadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="JPSearch:schema:coremetadata jpcore.xsd">
  <Identifier>urn:unique:identifier:1:2:3
    </Identifier>
  <Modifiers>
    <GivenName>Jonh</GivenName>
    <FamilyName>Smith</FamilyName>
  </Modifiers>
</ImageDescription>
```

5.4 SourceType

5.4.1 Introduction

The `SourceType` type specifies the source of the image when the image is created from a pre-existing artificial work or product such as painting, book, video and so on.

5.4.2 Syntax

```
<complexType name="SourceType">
    <sequence>
      <element name="SourceElementType" type="string"/>
      <element name="SourceElement" maxOccurs="unbounded">
        <complexType>
          <sequence>
            <element name="SourceElementTitle" type="string"/>
            <element name="SourceElementDescription" type="string"/>
            <element name="SourceElementIdentifier" type="anyURI"
minOccurs="0"/>
            <!-- Editor's Note: the type of SourceElementIdentifier
needs to be eliminated if the image identifier type use another type
rather than anyURI -->
          </sequence>
        </complexType>
      </element>
      <element name="CreationMethod" type="string"/>
      <element name="CreationDescription" type="string"/>
    </sequence>
  </complexType>
  <complexType name="SourceElementType">
</complexType>
```

5.4.3 Semantic

Semantics of the `SourceType` type:

Name	Definition
`SourceType`	Specifies the `SourceType` type that is used for describing the source that is used for the image.
`SourceElementType`	Specifies a type of image source that can be an original image or an object in the form such as painting, book, image and so on.
`SourceElement~~Object~~`	-Describes the source element(s) that is used for creation of the image. In the case that the image is created by using multiple source elements, each source element is described in a SourceElement field.
`SourceElementTitle`	Specifies a title for the source element being used.

SourceElementDescription	Describes information for the source element that is used for creation of the image such as location, contact point, and so on.
SourceElementIdentifier	Specifies the identifier for the source element if exists.
CreationMethod	Specifies the method that is used for creation of the image. Recommended values for this field are Photographing, Film Scanning, Reflection Print Scanning, Video Still Capturing, and Computer Graphic Editing.
CreationDescription	Describes the information about the creation of the image. Or it can describe the technical information that is necessary for the production work of the image such as workflow parameters.

5.4.4 Example

This example shows the image created from an oil painting 'Monna Lisa' by photographing. The painting is created by 'Leonardo da Vinci' and is kept in 'Louvre, France'. The camera used for the production is 'CN 450D'.

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageDescription xmlns="JPSearch:schema:coremetadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="JPSearch:schema:coremetadata jpcore.xsd">
  <Identifier>urn:unique:identifier:1:2:3
    </Identifier>
  <Modifiers>
    <GivenName>Jonh</GivenName>
    <FamilyName>Smith</FamilyName>
  </Modifiers>
</ImageDescription>
```

5.5 PublisherType

5.5.1 Introduction

The `PublisherType` type supports the description of information about the publisher which can be any public or private organization or person.

5.5.2 Syntax

```
<complexType name="PublisherType">
  <sequence>
    <choice minOccurs="0" maxOccurs="unbounded">
      <element name="PersonName" type="JPCore:PersonNameType"/>
      <element name="OrganizationInformation"
type="JPCore:OrganizationType"/>
    </choice>
  </sequence>
</complexType>
```

5.5.3 Semantic

Semantics of the `PublisherType` type:

Name	Definition
`PublisherType`	Specifies information of the publisher for the image. No information can be defined. The publisher information can be either in the form of person's name or organization's description.
`PersonName`	Describes a publisher in the form of a person's name.
`OrganizationInformation`	Describes a publishing organization.

5.5.4 Example

This example describes the information given for the publisher of an image. It supports the annotation of an organization and/or a human being.

```
<ImageDescription>
    <?xml version="1.0" encoding="UTF-8"?>
<ImageDescription xmlns="JPSearch:schema:coremetadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="JPSearch:schema:coremetadata jpcore.xsd">
  <Identifier>urn:unique:identifier:1:2:3
    </Identifier>
  <Publisher>
    <PersonName>
      <GivenName>Jonh</GivenName>
      <FamilyName>Smith</FamilyName>
    </PersonName>
    <OrganizationInformation>
      <Name>University of Passau</Name>
      <Address>
        <Name>Innstrasse 43, 94034 Passau, Germany
                </Name>
        <Description>
                    Small city at the border to Austria surrounded
                    by three rivers
                </Description>
      </Address>
    </OrganizationInformation>
  </Publisher>
</ImageDescription>
```


5.6 RightsDescriptionType

5.6.1 Introduction

The RightsDescriptionType type is used to provide information about rights description standard and corresponding rights description. Among the information, a URN value is necessary for the existing rights description standard. In addition, free text to describe the standard might be provided. Actual rights description based on the specified standard would be provided using either reference external file or embedding the description itself as a string. Existence of the rights description itself might be optional.

5.6.2 Syntax

```
<complexType name="RightsDescriptionType">
  <sequence>
    <element        name="RightsDescriptionInformation"        type="anyURI"
minOccurs="1" maxOccurs="1"/>
    <element name="Description" type="string" minOccurs="0"/>
    <element      name="ActualRightsDescriptionReference"      type="anyURI"
minOccurs="1" maxOccurs="1"/>
    <element name="ActualRightsDescription" type="string" minOccurs="0"/>
  </sequence>
 </complexType>
```

5.6.3 Semantics

Semantics of the `RightsDescriptionType` type:

5.6.4 Example

Name	*Definition*
`RightsDescriptionInformation`	Specifies the location where the rights description standard is provided.
`Description`	Describes the rights description standard in the form of free text.
`ActualRightsDescriptionReference`	Specifies an external file that includes the actual rights description.
`ActualRightsDescription`	Specifies the actual rights description.

This example shows the description of a rights description providing the location where the actual rights description exists.

```
<ImageDescription>
    <Identifier>urn:unique:identifier:1:2:3
    </Identifier>
    <RightsDescriptionType>
       <RightsDescriptionInformation>
        http://www.rdstandard.org
       </RightsDescriptionInformation>
       <Description>
        Provides a standard for rights description.
       </Description>
       <ActualRightsDescriptionReference>
        http://www.rdstandard.org/particulaValueRD
       </ActualRightsDescriptionReference>
    </RightsDescriptionType>
</ImageDescription>
```

5.7 PlaceType

5.7.1 Introduction

The `PlaceType` type supports the description of locations that are shown in the image or certain regions thereof.

5.7.2 Syntax

```
<complexType name="PlaceType">
  <sequence>
    <element name="Name" type="string" minOccurs="0"
          maxOccurs="unbounded"/>
    <element name="Description" type="string" minOccurs="0"
          maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

5.7.3 Semantics

Semantics of the `PlaceType` type:

Name	*Definition*
`PlaceType`	Specifies the definition of a place description.
`Name`	Specifies the name of the place.
`Description`	Specifies a description about the place.

5.7.4 Example

This example shows the description of a certain location shown in the image. The `PlaceType` type is embedded in the `ContentDescription` tag which supports the annotation of region specific information.

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageDescription                                    xmlns="JPSearch:schema:coremetadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="JPSearch:schema:coremetadata jpcore.xsd">
  <RegionOfInterest>
    <RegionLocator>
      <Region dim="2"> 0 0 100 100</Region>
    </RegionLocator>
    <Description>A short description about the selected region
        </Description>
    <Keyword>plenary meeting</Keyword>
    <Title>plenary meeting</Title>
    <ContentDescription>
      <Place>
        <Name>meeting room xyz</Name>
        <Description>meeting room at the sardinia JPEG meeting
                </Description>
      </Place>
```

```
    </ContentDescription>
  </RegionOfInterest>
</ImageDescription>
```

5.8 PersonType

5.8.1 Introduction

The `PersonType` type describes information about a certain human being shown within an image or a certain region.

5.8.2 Syntax

```
<complexType name="PersonType">
 <sequence>
  <element name="Name" type="JPCore:PersonNameType"/>
  <element name="Affiliation" type="JPCore:OrganizationType"
           minOccurs="0" maxOccurs="unbounded"/>
  <element name="Address" type="JPCore:PlaceType" minOccurs="0"/>
  <element name="Description" type="string" minOccurs="0"/>
  <element name="Nationality" type="string" minOccurs="0"/>
 </sequence>
</complexType>
```

5.8.3 Semantics

Semantics of the `PersonType` type:

Name	*Definition*
`PersonType`	Specifies the definition of a person description.
`Name`	Specifies the name of the person.
`Affiliation`	Specifies a description about the affiliation of the person.
`Address`	Specifies the address of the person.
`Description`	Specifies a description of the person.
`Nationality`	Specifies the nationality of the person.

5.8.4 Example

The following example demonstrates the description of a human being. Information about human beings can be given within annotated regions by using the `ContentDescription` tag.

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageDescription xmlns="JPSearch:schema:coremetadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="JPSearch:schema:coremetadata jpcore.xsd">
  <Identifier>urn:unique:identifier:1:2:3
    </Identifier>
  <RegionOfInterest>
    <RegionLocator>
      <Region dim="2"> 0 0 100 100</Region>
    </RegionLocator>
    <ContentDescription>
      <Person>
        <Name>
          <GivenName>Jonh</GivenName>
          <FamilyName>Smith</FamilyName>
        </Name>
        <Affiliation>
          <Name/>
        </Affiliation>
        <Address>
          <Name>Innstrasse 43, 94034 Passau, Germany
                    </Name>
          <Description>Small city at the border to Austria
                                        surrounded by three rivers
                    </Description>
        </Address>
        <Description>this is a description of the person
                </Description>
        <Nationality>AUT</Nationality>
      </Person>
    </ContentDescription>
  </RegionOfInterest>
</ImageDescription>
```


5.9 OrganizationType

5.9.1 Introduction

The `OrganizationType` type describes information about a certain organization (e.g., political, cultural, industry, etc.).

5.9.2 Syntax

```
<complexType name="OrganizationType">
  <sequence>
    <element name="Name" type="string" maxOccurs="unbounded"/>
    <element name="Address" type="JPCore:PlaceType"
             minOccurs="0"/>
  </sequence>
</complexType>
```

5.9.3 Semantics

Semantics of the `OrganizationType` type:

Name	*Definition*
`OrganizationType`	Specifies the definition of an organization description.
`Name`	Specifies the name of the organization.
`Address`	Specifies the address of the organization.

5.9.4 Example

The `OrganizationType` type supports the description of information about public, industrial, private, … organizations. The given example demonstrates its use within the `Publisher` tag.

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageDescription                                    xmlns="JPSearch:schema:coremetadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="JPSearch:schema:coremetadata jpcore.xsd">
  <Identifier>urn:unique:identifier:1:2:3
    </Identifier>
  <Publisher>
    <OrganizationInformation>
      <Name>University of Passau</Name>
      <Address>
        <Name>Innstrasse 43, 94034 Passau, Germany
                </Name>
        <Description>Small city at the border to Austria
                                        surrounded by three rivers
                </Description>
      </Address>
    </OrganizationInformation>
  </Publisher>
</ImageDescription>
```

5.10 EventType

5.10.1 Introduction

The `EventType` type describes information about a certain event (e.g., handshake, etc.) within a specific region of the image.

5.10.2 Syntax

```
<complexType name="EventType">
 <sequence>
  <element name="Label" type="anyURI" maxOccurs="unbounded"/>
  <element name="Description" type="string" minOccurs="0"
           maxOccurs="unbounded"/>
 </sequence>
</complexType>
```

5.10.3 Semantics

Semantics of the `EventType` type:

Name	*Definition*
`EventType`	Specifies the definition of an event description.
`Label`	Specifies the label of the event. A label is a concrete event instantiation (e.g., handshake) based on a specific taxonomy.
`Description`	Specifies the description of a label.

5.10.4 Example

The `EventType` type supports the annotation of events (e.g., handshaking, playing cards, etc.) that are shown in an image or a specific region thereof. An event is described by a `Label` and a `Description` tag. The `Label` tag refers to a URI specifying the event by a taxonomy. In series, the `Description` tag can provide additional information.

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageDescription xmlns="JPSearch:schema:coremetadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="JPSearch:schema:coremetadata jpcore.xsd">
  <Identifier>urn:unique:identifier:1:2:3</Identifier>
  <RegionOfInterest>
    <RegionLocator>
      <Region dim="2"> 0 0 100 100</Region>
    </RegionLocator>
    <Description>A short description about the selected region
        </Description>
    <Keyword>plenary meeting</Keyword>
    <Title>plenary meeting</Title>
    <ContentDescription>
      <Event>
        <Label>urn:writing:event:1:2</Label>
        <Description>writing some document
                </Description>
      </Event>
    </ContentDescription>
  </RegionOfInterest>
</ImageDescription>
```

5.11 ObjectType

5.11.1 Introduction

The `ObjectType` type describes information about a certain object (e.g., car, house, etc.) visible in an image or a certain region thereof.

5.11.2 Syntax

```
<complexType name="ObjectType" >
 <sequence>
  <element name="Name" type="string" maxOccurs="unbounded"/>
  <element name="Description" type="string" minOccurs="0" />
  <element name="Label" type="anyURI" minOccurs="0"/>
 </sequence>
</complexType>
```

5.11.3 Semantics

Semantics of the `ObjectType` type:

Name	*Definition*
`ObjectType`	Specifies the definition of an object (e.g., car, house) visible in the image.
`Name`	Specifies the name of the object.
`Description`	Specifies the description of the object.
`Label`	Specifies the label information of the object based on a defined taxonomy.

5.11.4 Example

The `ObjectType` type allows the annotation of arbitrary objects (car, house, table, etc.) shown at a certain region within an image. Similar to the `EventType` type, an object is identified by a `Label` tag pointing to a URI of a certain taxonomy. In addition, the name and a description of the object can be given as demonstrated in the example below.

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageDescription                                    xmlns="JPSearch:schema:coremetadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="JPSearch:schema:coremetadata jpcore.xsd">
  <Identifier>urn:unique:identifier:1:2:3</Identifier>
  <RegionOfInterest>
    <RegionLocator>
      <Region dim="2"> 0 0 100 100</Region>
    </RegionLocator>
    <Description>A short description about the selected region
      </Description>
    <Keyword>plenary meeting</Keyword>
    <Title>plenary meeting</Title>
    <ContentDescription>
      <Object>
        <Name>Laptop</Name>
        <Description>running laptop of UoP
                </Description>
        <Label>urn:laptop:id:1:2:3</Label>
      </Object>
    </ContentDescription>
  </RegionOfInterest>
</ImageDescription>
```

5.12 RegionOfInterestType

5.12.1 Introduction

The `RegionOfInterestType` type describes information about a certain region in an image.

5.12.2 Syntax

```
<complexType name="RegionOfInterestType">
  <sequence>
    <element name="RegionLocator" type="JPCore:RegionLocatorType"/>
    <element name="Description" type="string" minOccurs="0"/>
    <element name="Keyword" type="string" minOccurs="0"
             maxOccurs="unbounded"/>
    <element name="Title" type="string" minOccurs="0"/>
    <element name="ContentDescription" minOccurs="0">
      <complexType>
        <sequence>
          <element name="Person" type="JPCore:PersonType"
                   minOccurs="0" maxOccurs="unbounded"/>
          <element name="Object" type="JPCore:ObjectType"
                   minOccurs="0" maxOccurs="unbounded"/>
          <element name="Place" type="JPCore:PlaceType"
                   minOccurs="0" maxOccurs="unbounded"/>
          <element name="Event" type="JPCore:EventType"
                   minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
      </complexType>
    </element>
    <element name="ExternalDescription"
             type="JPCore:ExternalDescriptonType" minOccurs="0"
             maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

5.12.3 Semantics

Semantics of the `RegionOfInterestType` type:

Name	*Definition*
`RegionOfInterestType`	Specifies the definition of a certain region in the image.
`RegionLocator`	Specifies the spatial location of the region within the image.
`Description`	Specifies the description of the region.
`Keyword`	Specifies keywords for the region.
`Title`	Specifies a title for the region.
`ContentDescription`	Specifies the semantic description of the content shown within the region.
`Person`	Specifies information about visible persons in the region.
`Object`	Specifies information about visible objects in the region
`Place`	Specifies information about the location shown in the region
`Event`	Specifies information about events.
`ExternalDescription`	Specifies a formal way in order to describe information which is not covered by the core schema and can be integrated from other standards (e.g., MPEG-7). Note, by using this tag it is not guaranteed that the referenced description can be evaluated by all JPSearch compliant systems.

5.12.4 Example

The `RegionOfInterestType` type supports the description of regions within images. For such a region, a description, keywords and a title can be specified, Furthermore, the content of the region such as occurring persons, events, or shown places can be annotated by the `ContentDescription` tag. Descriptive information that is not covered by the core schema can be integrated by using either the `UserDefinedTag` tag or the `ExternalDescription` tag.

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageDescription                                xmlns="JPSearch:schema:coremetadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="JPSearch:schema:coremetadata jpcore.xsd">
  <Identifier>urn:unique:identifier:1:2:3</Identifier>
  <RegionOfInterest>
    <RegionLocator>
      <Region dim="2"> 0 0 100 100</Region>
    </RegionLocator>
    <Description>A short description about the selected region
        </Description>
```

```
    <Keyword>plenary meeting</Keyword>
    <Title>plenary meeting</Title>
    <ContentDescription>
      <Place>
        <Name>meeting room xyz</Name>
        <Description>meeting room at the sardinia JPEG meeting
                </Description>
      </Place>
    </ContentDescription>
    <ExternalDescription>
      <TagName                          fromNamespace="urn:mpeg:mpeg7:schema:2004"
fromNamespacePrefix="mpeg7">mpeg7:ColorLayout</TagName>
      <StructuredValue fromNamespace="urn:mpeg:mpeg7:schema:2004">
        <mpeg7:Mpeg7                       xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004"
xsi:schemaLocation="urn:mpeg:mpeg7:schema:2004 M7v2schema.xsd">
          <mpeg7:DescriptionUnit xsi:type="mpeg7:ColorLayoutType">
            <mpeg7:YDCCoeff>1</mpeg7:YDCCoeff>
            <mpeg7:CbDCCoeff>2</mpeg7:CbDCCoeff>
            <mpeg7:CrDCCoeff>3</mpeg7:CrDCCoeff>
            <mpeg7:YACCoeff2>1 2</mpeg7:YACCoeff2>
            <mpeg7:CbACCoeff2>1 2</mpeg7:CbACCoeff2>
            <mpeg7:CrACCoeff2>1 2</mpeg7:CrACCoeff2>
          </mpeg7:DescriptionUnit>
        </mpeg7:Mpeg7>
      </StructuredValue>
    </ExternalDescription>
  </RegionOfInterest>
</ImageDescription>
```

5.13 RegionLocatorType

5.13.1 Introduction

The `RegionLocatorType` type describes information about the location of a certain region in an image.

5.13.2 Syntax

```
<complexType name="RegionLocatorType">
  <sequence>
    <element name="Region" type="JPCore:IntegerMatrixType"
             maxOccurs="unbounded"/>
  </sequence>
</complexType>
<complexType name="IntegerMatrixType">
  <simpleContent>
    <extension base="JPCore:listOfInteger">
      <attribute name="dim" type="positiveInteger" use="required"/>
    </extension>
  </simpleContent>
</complexType>
<simpleType name="listOfInteger">
    <list itemType="int"/>
</simpleType>
```

5.13.3 Semantics

Semantics of the `RegionLocatorType` type:

Name	*Definition*
`RegionLocatorType`	Specifies the definition of the location of a certain region in the image.
`Region`	Specifies the spatial location and dimension of a region within the image.
`IntegerMatrixType`	Specifies a spatial region. A spatial region is defined by the `IntegerMatrixType` type which allows the specification of a list of positive integer values describing individual points. The amount of necessary integer values per point is defined by the `dim` attribute of `IntegerMatrixType` type. The individual points define the region where two points mean a rectangular, three points a triangle and so on.
`dim`	Attribute that defines the dimension of the individual points (e.g., 2 means that 2 integer values are required to define one point)
`listOfInteger`	Simple type that specifies a list of positive integer values which describes individual points. The amount of necessary integer values per point is defined by the `dim` attribute of the `IntegerMatrixType` type.

5.13.4 Example

The RegionLocatorType type allows the specification of regions within images. Every region can be of arbitrary shape depending on the used dimension and amount of points. For instance the given example describes a two dimensional (the dim attribute is set to 2) region (a rectangular) by the following coordinates (0, 0), (100, 100).

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageDescription xmlns="JPSearch:schema:coremetadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="JPSearch:schema:coremetadata jpcore.xsd">
  <Identifier>urn:unique:identifier:1:2:3</Identifier>
  <RegionOfInterest>
    <RegionLocator>
      <Region dim="2"> 0 0 100 100</Region>
    </RegionLocator>
  </RegionOfInterest>
</ImageDescription>
```

5.14 ExternalDescriptonType

5.14.1 Introduction

The `ExternalDescriptonType` type allows embedding metadata fields not defined by JPEG but used in external user communities or defined by external metadata formats (e.g., MPEG-7 descriptors or descriptor schemes).

5.14.2 Syntax

```
<complexType name="ExternalDescriptonType">
   <sequence>
     <element name="TagName">
       <complexType mixed="true">
         <attribute name="fromNamespace" type="anyURI"
                    use="optional"/>
         <attribute name="fromNamespacePrefix" type="string"
                    use="optional"/>
       </complexType>
     </element>
     <choice>
       <element name="LiteralValue">
         <complexType mixed="true">
           <attribute name="fromControlledVocabularyOfLiterals"
                      type="anyURI" use="optional"/>
           <attribute name="fromGrammarOfLiterals" type="anyURI"
                      use="optional"/>
         </complexType>
       </element>
       <element name="ResourceValue">
         <complexType mixed="true">
           <attribute name="fromNamespace" type="anyURI"
                      use="optional"/>
         </complexType>
       </element>
      <element name="StructuredValue">
        <complexType mixed="true">
          <sequence>
            <any namespace="##any"/>
          </sequence>
          <attribute name="fromNamespace" type="anyURI"
use="required"/>
        </complexType>
      </element>
     </choice>
   </sequence>
 </complexType>
```

5.14.3 Semantics

Semantics of the `ExternalDescriptonType` type:

Name	*Definition*
`ExternalDescriptonType`	Allows the integration of components of other metadata formats (e.g., MPEG-7 descriptors or descriptor schemes). It can be 1) a flat literal value, 2) a resource value (identified by a URI) or 3) a structured XML value from an external namespace.
`TagName`	Specifies the name of the tag.
`fromNamespace`	Optional attribute which specifies a namespace to which the tag name belongs.
`fromNamespacePrefix`	Optional attribute which specifies the prefix of the namespace to which the tag belongs.
`LiteralValue`	Specifies a literal value for the tag.
`fromControlledVocabularyOfLiterals`	Optional attribute which specifies, if there exists one, the URI of a controlled vocabulary to which the literal value belongs (e.g. MIME types)
`fromGrammarOfLiterals`	Optional attribute which specifies, if there exists one, the URI of a grammar of literals to which the literal value belongs (e.g. a datetime format)
`ResourceValue`	Specifies the URI pointing to a resource.
`fromNamespace`	Optional attribute which specifies, if there exists one, the URI of a namespace to which the resource value belongs.
`StructuredValue`	Specifies a structured XML value from an external namespace.
`fromNamespace`	Optional attribute which specifies, if there exists one, the URI of a namespace to which the description belongs.

5.14.4 Example

The following example shows a UserDefinedTag being used to wrap an Mpeg7 description.

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageDescription                                    xmlns="JPSearch:schema:coremetadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="JPSearch:schema:coremetadata jpcore.xsd">
  <Identifier>urn:unique:identifier:1:2:3</Identifier>
  <RegionOfInterest>
    <RegionLocator>
      <Region dim="2"> 0 0 100 100</Region>
    </RegionLocator>
    <ExternalDescription>
      <TagName                              fromNamespace="urn:mpeg:mpeg7:schema:2004"
fromNamespacePrefix="mpeg7">mpeg7:ColorLayout</TagName>
      <StructuredValue fromNamespace="urn:mpeg:mpeg7:schema:2004">
        <mpeg7:Mpeg7                          xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004"
```

```
xsi:schemaLocation="urn:mpeg:mpeg7:schema:2004 M7v2schema.xsd">
          <mpeg7:DescriptionUnit xsi:type="mpeg7:ColorLayoutType">
            <mpeg7:YDCCoeff>1</mpeg7:YDCCoeff>
            <mpeg7:CbDCCoeff>2</mpeg7:CbDCCoeff>
            <mpeg7:CrDCCoeff>3</mpeg7:CrDCCoeff>
            <mpeg7:YACCoeff2>1 2</mpeg7:YACCoeff2>
            <mpeg7:CbACCoeff2>1 2</mpeg7:CbACCoeff2>
            <mpeg7:CrACCoeff2>1 2</mpeg7:CrACCoeff2>
          </mpeg7:DescriptionUnit>
        </mpeg7:Mpeg7>
      </StructuredValue>
    </ExternalDescription>
    <ExternalDescription>
      <TagName                                          fromNamespace="urn:medical"
fromNamespacePrefix="mpeg7">PatientName</TagName>
      <LiteralValue>John Smith</LiteralValue>
    </ExternalDescription>
    <ExternalDescription>
      <TagName                                          fromNamespace="urn:medical"
fromNamespacePrefix="mpeg7">PatientName</TagName>
      <ResourceValue>http://www.patients.org/jsmith</ResourceValue>
    </ExternalDescription>
  </RegionOfInterest>
</ImageDescription>
```

5.15 ControlledRatingTermType

5.15.1 Introduction

The `ControlledRatingTermType` type specifies rating value and the definition of terms used for image rating. The terms for rating should be one of controlled terms that are provided by JPSearch.

5.15.2 Syntax

```
<complexType name="ControlledRatingTermType">
  <sequence>
    <element name="LabelDefinition" type="anyURI"/>
    <element name="LabelValue" type="string"/>
  </sequence>
</complexType>
```

5.15.3 Semantics

Semantics of the `ControlledRatingTermType` type:

Name	*Definition*
`ControlledRatingTermType`	Specifies the `ControlledRatingTermType` type that is used for rating and corresponding term definition.
`LabelDescription`	Specifies the location of the term definition.
`LabelValue`	Specifies the rating value for the image. The rating value should be defined in the location given in LavelDefinition.

5.15.4 Example

This example shows the image has rating label ‘General’ that is defined in ‘urn:unique:ratingdefinition:1’.

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageDescription                                xmlns="JPSearch:schema:coremetadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="JPSearch:schema:coremetadata jpcore.xsd">
  <Identifier>urn:unique:identifier:1:2:3
  </Identifier>
  <Rating>
    <LabelDefinition>urn:ratingdefinition:1
     </LabelDefinition>
    <LabelValue>General</LabelValue>
  </Rating>
</ImageDescription>
```

5.16 ImageIdentifierType

5.16.1 Introduction

The `ImageIdentifierType` type provides the necessary information so that the origination of a particular image can be described. Moreover, identifiers that are created by a particular organization or method can be provided.

5.16.2 Syntax

```
<complexType name="ImageIdentifierType">
   <sequence>
     <element name="OriginationOfID" type="anyURI"/>
     <element name="Identifier" type="string"/>
   </sequence>
 </complexType>
```

5.16.3 Semantics

Semantics of the `ImageIdentifierType` type:

Name	Definition
`OriginationOfID`	Specifies either the organization or person who issued the identifier, or the location where the method used to create the identifier can be found.
`Identifier`	Specifies the identifier issued by the particular organization or the method specified by the `OriginationOfID`.

5.16.4 Example

This example shows the usage of OriginalImageIdentifier in order to describe identifier that is created by a particular user.

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageDescription xmlns="JPSearch:schema:coremetadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="JPSearch:schema:coremetadata jpcore.xsd">
  <Identifier>
urn:unique:identifier:1:2:3
    </Identifier>
  <OriginalImageIdentifier>
    <OriginationOfID>
http::www.newidentifier.com/ID3_Algorithm
</OriginationOfID>
    <Identifier>
          Id3::ABCED1234ABCDFFCD
</Identifier>
  </OriginalImageIdentifier>
</ImageDescription>
```

5.17 GPSPositioningType

5.17.1 Introduction

The GPSPositioningType type provides information in order to describe the location of a certain place.

5.17.2 Syntax

```
<complexType name="GPSPositioningType">
  <attribute name="longitude" use="required">
    <simpleType>
      <restriction base="double">
        <minInclusive value="-180.0"/>
        <maxInclusive value="180.0"/>
      </restriction>
    </simpleType>
  </attribute>
  <attribute name="latitude" use="required">
    <simpleType>
      <restriction base="double">
        <minInclusive value="-90.0"/>
        <maxInclusive value="90.0"/>
      </restriction>
    </simpleType>
  </attribute>
  <attribute name="altitude" type="double" use="optional"/>
</complexType>
```

5.17.3 Semantics

Semantics of the `GPSPositioningType` type:

Name	*Definition*
`GPSPositioningType`	Specifies the geographic position of a place shown within the image.
`longitude`	Specifies the longitude of the place.
`latitude`	Specifies the latitude of the place
`altitude`	Specifies the altitude of the place.

5.17.4 Example

This example shows the use of GPS coordinates in order to describe location information.

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageDescription xmlns="JPSearch:schema:coremetadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="JPSearch:schema:coremetadata jpcore.xsd">
  <Identifier>urn:unique:identifier:1:2:3</Identifier>
  <GPSPositioning latitude="34" longitude="34" altitude="10"/>
</ImageDescription>
```

6 Management of core schema and translation rules

6.1 General

This Clause targets on the definition of the complex types for the registration process of the schema, its translation rules and contact information. The process of registration is mandatory for all schemas that can be addressed within a retrieval operation. The detailed process of registration can be found in Annex A. The standard supports two scenarios. First, a global authority for schemas and their translation rules will be established where all JPSearch compliant retrieval applications can obtain the needed information. The list of registration authorities shall be listed at http://www.iso.org/iso/maintenance_agencies. Second, in case the retrieval application operates in offline mode, the schema and their translation rules shall be located at the application itself.

6.2 Wrapper of the schema

The description examples and syntax of description tools specified in this document assume that a schema wrapper is provided which identifies the XML Schema namespace (XML Schema) and JPSearch namespace for the management part:

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
            xmlns:xs="http://www.w3.org/2001/XMLSchema"
            xmlns:jps_mngt="JPSearch:schema:management"
            xmlns:jpqf="urn:jpeg:jpqf:schema:2008"
            targetNamespace="JPSearch:schema:management"
            elementFormDefault="qualified">
   <import namespace="urn:jpeg:jpqf:schema:2008"
            schemaLocation="JPQF_SFCD.xsd"/>
```

The following tag is used to close the schema:

```
</schema>
```

6.3 Root Element

6.3.1 Introduction

The `SchemaManagement` element serves as the root element of the JPSearch Management Process. The root element shall be used as the topmost element in all messages transmitted. This applies on all operations in the corresponding message direction (input or output).

6.3.2 Syntax

```
<element name="SchemaManagement" type="jps_mngt:ManagementType"/>

<complexType name="ManagementType">
  <choice>
 <element name="Register">
   <complexType>
    <choice>
      <element name="Input" type="jps_mngt:RegisterInputType"/>
      <element name="Output" type="jps_mngt:RegisterOutputType"/>
    </choice>
   </complexType>
 </element>
 <element name="Request">
   <complexType>
    <choice>
      <element name="Input" type="jps_mngt:RequestInputType"/>
      <element name="Output" type="jps_mngt:RequestOutputType"/>
    </choice>
   </complexType>
 </element>
 <element name="Replace">
   <complexType>
    <choice>
      <element name="Input" type="jps_mngt:ReplaceInputType" />
      <element name="Output" type="jps_mngt:ReplaceOutputType" />
    </choice>
   </complexType>
 </element>
  </choice>
  <attribute name="jpsearchID" type="anyURI" use="required"/>
 </complexType>
```

6.3.3 Semantics

Semantics of the `ManagementType` type:

Name	*Definition*
SchemaManagement	Serves as the root element of the JPSearch Management Process. The SchemaManagement element shall be used as the topmost element in any JPsearch management message.
ManagementType	Specifies the syntax of the root element. Within this element, either one of the following operations can be chosen: Register, Request and Replace.
Register	Allows the registration of a schema, its translation rules and additional information.
Input	Wraps the user register request specified using the Register Input Format, which is to be sent from the service provider to the JPSearch schema authority.
Output	Wraps the result (e.g., error message, ack, etc.) of the register request specified using the Register Output Format.
Request	Allows the request of registered schema information and translation rules.
Input	Specifies an input request for a schema or translation rules
Output	Specifies the output of a request for a schema or translation rules
Replace	Specifies the replacement functionality of schema related information such as provider information and translation rules.
Input	Specifies the input of a replacement request.
Output	Specifies the output of a replacement request.
jpsearchID	Specifies a unique identifier which is assigned to every message sent between a requester and a responder.

6.3.4 Example

This example shows a number of cases according to different schema management operations. The first example shows a scenario in which a new schema 'DC-Based-Schema which is designed by 'Schema Creator' is being registered with two translation rules.

In the following examples, in order to simplify the code, the namespace is assumed as 'urn:JPSearch jps_mngt'.

```
<SchemaManagement jpsearchID= "JPS-MSG-01"
                  xmlns="JPSearch:schema:management">
   <Register>
      <Input>
         <ProviderInformation>
           <Name>Schema Creator</Name>
           <Description>Image schema Developer based on
MPEG</Description>
           <Contact>
              <EMail>creator-email@schema.com</EMail>
              <PostalAddress> 2 1st St., Los Angeles, CA 876554,
USA</PostalAddress>
```

```
            <Phone>212-1234-4567</Phone>
            <lang>eng</lang>
         </Contact>
            <ProviderCapability xsi:type="QueryCapabilityType">
              <ServiceCapability>
                <jpqf:SupportedQueryTypes href="urn:X:Y:Z"/>
              </ServiceCapability>
            </ProviderCapability>
            <ProviderCapability xsi:type="BenchmarkCapabilityType"
                                name="Example Benchmark"
                                reference="urn:unique_identifier">
                <QualityRating queryType="urn:X:Y:Z" rating="0.3"/>
            </ProviderCapability>
      </ProviderInformation>
      <Schema>
         <SchemaInformation>
            <Description>new version 2009</Description>
            <SpecificationName>DC-Based-Schema</SpecificationName>
            <Version>1.0</Version>
            <Identifier>urn:DC-Based-Schema:1</Identifier>
            <Location>urn:DC-Based-Schema:registry:1</Location>
         </SchemaInformation>
         <TranslationRules>
            <TranslationRule>
                <!-- example of one-to-one translation -->
                <FromField>DCB-Field 1</FromField>
                <ToField>Core-Field K</ToField>
            </TranslationRule>
            <TranslationRule>
                <!-- example of one-to-one translation -->
                <FromField>DCB-Field 2</FromField>
                <ToField>Core-Field K+1</ToField>
            </TranslationRule>
            <fromFormat>urn:DC-Based-Schema:1</fromFormat>
            <toFormat>urn:JPSearchCore:1</toFormat>
         </TranslationRules>
      </Schema>
   </Input>
</Register>
</SchemaManagement>
```

The second example shows a result when a new schema `'DC-Based-Schema'` is registered with the appropriate translation rules.

```
<SchemaManagement jpsearchID="JPS-MSG-02"
xmlns="JPSearch:schema:management">
   <Register>
      <Output>
         <Identifier>urn:DC-Based-Schema:1</Identifier>
         <Status>
            <Code>status-code-x</Code>
            <Description>registered and being evaluated</Description>
         </Status>
      </Output>
   </Register>
</SchemaManagement>
```

The third example shows a scenario when schema information is requested.

```
<SchemaManagement jpsearchID="JPS-MSG-03"
                  xmlns="JPSearch:schema:management">
   <Request>
      <Input>
         <SchemaID>urn:DC-Based-Schema:1</SchemaID>
      </Input>
   </Request>
</SchemaManagement>
```

The forth example shows a output message for a request input message.

```
<SchemaManagement jpsearchID="JPS-MSG-04"
                  xmlns="JPSearch:schema:management">
   <Request>
      <Output>
         <ResultItem>
            <Identifier>urn:DC-Based-Schema:1</Identifier>
            <Location>urn:DC-Based-Schema:registry:1</Location>
            <ProviderInformation>
              <Name>Schema Creator</Name>
              <Description>Image schema Developer based on MPEG
              </Description>
              <Contact>
                 <EMail>creator-email@schema.com</EMail>
                 <PostalAddress> 2 1st St., Los Angeles, CA 876554, USA
                  </PostalAddress>
                 <Phone>212-1234-4567</Phone>
                 <lang>eng</lang>
              </Contact>
              <ProviderCapability xsi:type="QueryCapabilityType">
                <ServiceCapability>
                  <jpqf:SupportedQueryTypes href="urn:X:Y:Z"/>
                </ServiceCapability>
              </ProviderCapability>
              <ProviderCapability xsi:type="BenchmarkCapabilityType"
                                  name="Example Benchmark"
                                  reference="urn:unique_identifier">
                  <QualityRating queryType="urn:X:Y:Z" rating="0.3"/>
              </ProviderCapability>
            </ProviderInformation>
         </ResultItem>
      </Output>
   </Request>
</SchemaManagement>
```

The fifth example shows an input scenario for replace operation that changes a translation rule.

```
<SchemaManagement jpsearchID="JPS-MSG-05"
                  xmlns="JPSearch:schema:management">
   <Replace>
      <Input>
         <SchemaID>urn:DC-Based-Schema:1</SchemaID>
         <TranslationRules>
              <TranslationRule>
```

```
                    <!-- example of one-to-one translation -->
                    <FromField>DCB-Field 1</FromField>
                    <ToField>Core-Field K+3</ToField>
                </TranslationRule>
                <fromFormat>urn:DC-Based-Schema:1</fromFormat>
                <toFormat>urn:JPSearchCore:1</toFormat>
            </TranslationRules>
        </Input>
    </Replace>
</SchemaManagement>
```

The sixth example shows an output scenario for the above input message.

```
<SchemaManagement jpsearchID="JPS-MSG-06"
                  xmlns="JPSearch:schema:management">
    <Replace>
        <Output>
            <Identifier>urn:DC-Based-Schema:1</Identifier>
            <Status>
                <Code>status-code-y</Code>
                <Description>new trans. Rule registered and being evaluated
                </Description>
            </Status>
        </output>
    </Replace>
</SchemaManagement>
```


6.4 RegisterInputType

6.4.1 Introduction

The `RegisterInputType` type provides information that is required during the registration process of schema and translation rules.

6.4.2 Syntax

```
<complexType name="RegisterInputType">
    <sequence>
        <element name="ProviderInformation"
        type="jps_mngt:ProviderInformationType"/>
         <element name="Schema" type="jps_mngt:SchemaType"
minOccurs="0"/>
    </sequence>
</complexType>
```

6.4.3 Semantics

Semantics of the `RegisterInputType` type:

Name	*Definition*
`RegisterInputType`	Specifies the RegisterInputType type which is used for describing all information that is necessary during the registration of schemas and translation rules.
`ProviderInformation`	Specifies the information that identifies the provider of the schema.
`Schema`	Specifies details about the registered schema and its translation rules

6.4.4 Example

The following example shows a scenario in which a new schema `'DC-Based-Schema'` which is designed by `'Schema Creator'` is being registered with a translation rule.

```
<SchemaManagement jpsearchID="JPS-MSG-01"
                  xmlns="JPSearch:schema:management">
   <Register>
      <Input>
         <ProviderInformation>
           <Name>Schema Creator</Name>
           <Description>Image schema Developer based on
MPEG</Description>
           <Contact>
              <EMail>creator-email@schema.com</EMail>
              <PostalAddress>2 1st St., Los Angeles, CA 876554, USA
              </PostalAddress>
              <Phone>212-1234-4567</Phone>
              <lang>eng</lang>
           </Contact>
              <ProviderCapability xsi:type="QueryCapabilityType">
                <ServiceCapability>
                  <jpqf:SupportedQueryTypes href="urn:X:Y:Z"/>
                </ServiceCapability>
              </ProviderCapability>
              <ProviderCapability xsi:type="BenchmarkCapabilityType"
                                  name="Example Benchmark"
                                  reference="urn:unique_identifier">
                  <QualityRating queryType="urn:X:Y:Z" rating="0.3"/>
              </ProviderCapability>
        </ProviderInformation>
        <Schema>
           <SchemaInformation>
              <Description>new version 2009</Description>
              <SpecificationName>DC-Based-Schema</SpecificationName>
              <Version>1.0</Version>
              <Identifier>urn:DC-Based-Schema:1</Identifier>
              <Location>urn:DC-Based-Schema:registry:1</Location>
           </SchemaInformation>
           <TranslationRules>
              <TranslationRule>
```

```
                    <!-- example of one-to-one translation -->
                    <FromField>DCB-Field 1</FromField>
                    <ToField>Core-Field K</ToField>
                </TranslationRule>
                <fromFormat>urn:DC-Based-Schema:1</fromFormat>
                <toFormat>urn:JPSearchCore:1</toFormat>
            </TranslationRules>
        </Schema>
    </Input>
  </Register>
</SchemaManagement>
```

6.5 RequestInputType

6.5.1 Introduction

The RequestInputType type allows the formulation of requests of registered schema and translation rules.

6.5.2 Syntax

```
<complexType name="RequestInputType" >
  <choice>
   <element name="SchemaID" type="anyURI" />
   <element name="TranslationRuleFrom" type="anyURI" />
   <element name="TranslationRuleTo" type="anyURI" />
  </choice>
</complexType>
```

6.5.3 Semantics

Semantics of the `RequestInputType` type:

Name	*Definition*
`RequestInputType`	Specifies the `RequestInputType` type which is used for describing all information that is necessary during the request of schemas and/or translation rules.
`SchemaID`	Requests the schema information that is registered by the specified URI.
`TranslationRuleFrom`	Requests the information for the translation rules that are registered for the translation between a proprietary schema to the core schema. The proprietary schema is identified by its schema ID.
`TranslationRuleTo`	Requests the information for the translation rules that are registered for the translation between the core schema to a proprietary schema. The proprietary schema is identified by its schema ID.

6.5.4 Example

The following example shows a message that requests information when the source schema for translation is given.

```
<SchemaManagement jpsearchID="JPS-MSG-07"
                  xmlns="JPSearch:schema:management">
   <Request>
      <Input>
         <TranslationRuleFrom>urn:DC-Based-Schema:1
         </TranslationRuleFrom>
      </Input>
   </Request>
</SchemaManagement>
```

6.6 RequestOutputType

6.6.1 Introduction

The `RequestOutputType` type specifies the result format for a schema or translation rule request.

6.6.2 Syntax

```
<complexType name="RequestOutputType" >
  <sequence>
   <element name="ResultItem" maxOccurs="unbounded">
    <complexType>
      <sequence>
      <element name="Identifier" type="anyURI" />
      <element name="Location" type="anyURI" maxOccurs="unbounded" />
      <element name="ProviderInformation"
                        type="jps_mngt:ProviderInformationType" />
      </sequence>
    </complexType>
   </element>
  </sequence>
</complexType>
```

6.6.3 Semantics

Semantics of the `RequestOutputType` type:

Name	*Definition*
`RequestOutputType`	Specifies the `RequestOutputType` type which describes the information that is returned for an input request.
`ResultItem`	Specifies the container for one piece of information that fits the incoming request.
`Identifier`	Specifies the ID of either the schema or the translation rules that have been requested.
`Location`	Specifies the location where the schema or the translation rules can be obtained.
`ProviderInformation`	Specifies the provider information which corresponds to the requested schema or translation rules.

6.6.4 Example

The following example shows a resulting output message for a request input message.

```
<SchemaManagement jpsearchID="JPS-MSG-04"
                  xmlns="JPSearch:schema:management">
   <Request>
      <Output>
         <ResultItem>
            <Identifier>urn:DC-Based-Schema:1</Identifier>
            <Location>urn:DC-Based-Schema:registry:1</Location>
            <ProviderInformation>
              <Name>Schema Creator</Name>
              <Description>Image schema Developer based on MPEG
              </Description>
              <Contact>
                 <EMail>creator-email@schema.com</EMail>
                 <PostalAddress> 2 1st St., Los Angeles, CA 876554, USA
                  </PostalAddress>
                 <Phone>212-1234-4567</Phone>
                 <lang>eng</lang>
              </Contact>
              <ProviderCapability xsi:type="QueryCapabilityType">
                <ServiceCapability>
                  <jpqf:SupportedQueryTypes href="urn:X:Y:Z"/>
                </ServiceCapability>
              </ProviderCapability>
              <ProviderCapability xsi:type="BenchmarkCapabilityType"
                                  name="Example Benchmark"
                                  reference="urn:unique_identifier">
                  <QualityRating queryType="urn:X:Y:Z" rating="0.3"/>
              </ProviderCapability>
            </ProviderInformation>
         </ResultItem>
      </Output>
   </Request>
</SchemaManagement>
```


6.7 ProviderInformationType

6.7.1 Introduction

The `ProviderInformationType` type provides information in order to identify a certain provider.

6.7.2 Syntax

```
<complexType name="ProviderInformationType">
    <sequence>
        <element name="Name" type="string"/>
        <element name="Description" type="string" minOccurs="0"/>
        <element name="Contact" type="jps_mngt:ContactType"/>
        <element name="ProviderCapability"
             type="jps_mngt:ProviderCapabilityType" minOccurs="2"
```

```
            maxOccurs="unbounded"/>
    </sequence>
</complexType>

<complexType name="ProviderCapabilityType" abstract="true">
</complexType>
```

6.7.3 Syntax

Name	*Definition*
`ProviderInformationType`	Specifies a `ProviderInformationType` type which is used for describing all information about a certain provider.
`Name`	Specifies the name of a provider.
`Description`	Specifies details about a provider.
`Contact`	Specifies contact information of a provider.
`ProviderCapability`	Specifies a placeholder for capability description of the provider. The description contains query capabilities and benchmark capabilities. In addition, the standard is open to integrate domain specific capabilities. All introduced capability types shall inherit from the abstract `ProviderCapabilityType` type. As the `ProviderCapabilityType` type is abstract and extended by several subtypes, the `QueryCapabilityType` type and the `BenchmarkCapabilityType` type shall be present in every instance document.

6.7.4 Example

In the following example, the usage of `ProviderInformationType` is demonstrated within the element ProviderInformation.

```
<SchemaManagement jpsearchID="JPS-MSG-04"
                  xmlns="JPSearch:schema:management">
   <Request>
      <Output>
         <ResultItem>
            <Identifier>urn:DC-Based-Schema:1</Identifier>
            <Location>urn:DC-Based-Schema:registry:1</Location>
            <ProviderInformation>
              <Name>Schema Creator</Name>
              <Description>Image schema Developer based on MPEG
              </Description>
              <Contact>
                 <EMail>creator-email@schema.com</EMail>
                 <PostalAddress> 2 1st St., Los Angeles, CA 876554, USA
                  </PostalAddress>
```

```
                <Phone>212-1234-4567</Phone>
                <lang>eng</lang>
              </Contact>
              <ProviderCapability xsi:type="QueryCapabilityType">
                <ServiceCapability>
                  <jpqf:SupportedQueryTypes href="urn:X:Y:Z"/>
                </ServiceCapability>
              </ProviderCapability>
              <ProviderCapability xsi:type="BenchmarkCapabilityType"
                                  name="Example Benchmark"
                                  reference="urn:unique_identifier">
                <QualityRating queryType="urn:X:Y:Z" rating="0.3"/>
              </ProviderCapability>
            </ProviderInformation>
          </ResultItem>
        </Output>
    </Request>
</SchemaManagement>
```

6.8 ContactType

6.8.1 Introduction

The `ContactType` type provides information in order to contact a certain provider.

6.8.2 Syntax

```
<complexType name="ContactType">
    <sequence>
        <element name="EMail" type="string"/>
        <element name="PostalAddress" type="string" minOccurs="0"/>
        <element name="Phone" type="string" minOccurs="0"/>
    </sequence>
    <attribute ref="xml:lang" use="optional"/>
</complexType>
```

6.8.3 Semantics

Semantics of the `ContactType` type:

Name	*Definition*
`ContactType`	Specifies the `ContactType` type which is used for describing all necessary contact information about a certain provider.
`EMail`	Specifies the email address of a provider.
`PostalAddress`	Specifies the postal address about a provider.
`Phone`	Specifies the phone information of a provider.

6.8.4 Example

In the following example, the usage of `ContactType` is demonstrated within the element Contact.

```
<SchemaManagement jpsearchID="JPS-MSG-08"
                  xmlns="JPSearch:schema:management">
   <Register>
      <Input>
         <ProviderInformation>
           <Name>Schema Creator</Name>
           <Description>Image schema Developer based on
MPEG</Description>
           <Contact>
              <EMail>creator-email@schema.com</EMail>
              <PostalAddress> 2 1st St., Los Angeles, CA 876554, USA
              </PostalAddress>
              <Phone>212-1234-4567</Phone>
              <lang>eng</lang>
           </Contact>
              <ProviderCapability xsi:type="QueryCapabilityType">
                <ServiceCapability>
                  <jpqf:SupportedQueryTypes href="urn:X:Y:Z"/>
                </ServiceCapability>
              </ProviderCapability>
              <ProviderCapability xsi:type="BenchmarkCapabilityType"
                                  name="Example Benchmark"
                                  reference="urn:unique_identifier">
                  <QualityRating queryType="urn:X:Y:Z" rating="0.3"/>
              </ProviderCapability>
        </ProviderInformation>
      </Input>
   </Register>
</SchemaManagement>
```

6.9 QueryCapabilityType

6.9.1 Introduction

The `QueryCapabilityType` type extends the `ProviderCapabilityType` and describes the service capabilities of the provider. The service capability type allows the description of supported query types, supported metadata formats, etc. See the service capability description of the JPEG Query Format for detailed information.

6.9.2 Syntax

```
<complexType name="QueryCapabilityType">
  <complexContent>
    <extension base="jps_mngt:ProviderCapabilityType">
      <sequence>
        <element name="ServiceCapability"
                 type="jpqf:CapabilityType"/>
        </sequence>
    </extension>
  </complexContent>
</complexType>
```

6.9.3 Semantics

Semantics of the `QueryCapabilityType` type:

Name	*Definition*
`QueryCapabilityType`	Specifies the supported retrieval capabilities of the provider.
`ServiceCapability`	Specifies the retrieval capabilities of a provider. The service capability type allows the description of supported query types, supported metadata formats, etc. See the service capability description of the JPEG Query Fromat for detailed information.

6.9.4 Example

In the following example, the usage of `QueryCapabilityType` is demonstrated within the element ProviderInformation.

```
<SchemaManagement jpsearchID="JPS-MSG-04"
                  xmlns="JPSearch:schema:management">
   <Request>
      <Output>
         <ResultItem>
            <Identifier>urn:DC-Based-Schema:1</Identifier>
            <Location>urn:DC-Based-Schema:registry:1</Location>
            <ProviderInformation>
              <Name>Schema Creator</Name>
              <Description>Image schema Developer based on MPEG
              </Description>
              <Contact>
                 <EMail>creator-email@schema.com</EMail>
                 <PostalAddress>2 1st St., Los Angeles, CA 876554, USA
                  </PostalAddress>
                 <Phone>212-1234-4567</Phone>
                 <lang>eng</lang>
              </Contact>
              <ProviderCapability xsi:type="QueryCapabilityType">
                <ServiceCapability>
                  <jpqf:SupportedQueryTypes href="urn:X:Y:Z"/>
                </ServiceCapability>
              </ProviderCapability>
              <ProviderCapability xsi:type="BenchmarkCapabilityType"
                                  name="Example Benchmark"
                                  reference="urn:unique_identifier">
                  <QualityRating queryType="urn:X:Y:Z" rating="0.3"/>
              </ProviderCapability>
            </ProviderInformation>
         </ResultItem>
      </Output>
   </Request>
</SchemaManagement>
```

6.10 BenchmarkCapabilityType

6.10.1 Introduction

The `BenchmarkCapabilityType` type extends the `ProviderCapabilityType` and describes the evaluation results of a provider for its supported filter conditions. The particular type of benchmark being used is provided together with the rating achieved. For instance the retrieval quality of the provided search conditions such as QueryByMedia, TemporalQuery, etc. shall be defined. These measurements are being used to classify the service provided by the database accordingly. It allows for purposefully allocation of databases addressed corresponding user requirements.

6.10.2 Syntax

```
<complexType name="BenchmarkCapabilityType">
  <complexContent>
    <extension base="jps_mngt:ProviderCapabilityType">
      <sequence>
        <element name="QualityRating"
                 type="jps_mngt:QualityRatingType" minOccurs="1"
                 maxOccurs="unbounded"/>
      </sequence>
      <attribute name="name" type="string"/>
      <attribute name="reference" type="anyURI"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="QualityRatingType">
  <attribute name="queryType" type="anyURI"/>
  <attribute name="rating" type="float"/>
</complexType>
```

6.10.3 Semantics

Semantics of the `BenchmarkCapabilityType` type:

Name	*Definition*
`BenchmarkCapabilityType`	Specifies the benchmark description of a provider.
`QualityRating`	Specifies the quality rating for one filter condition of the service provider. This filter condition can be a QueryByMedia or TemporalQuery, etc. query type. See below for detailed information of the `QualityRatingType` type.
`name`	Specifies name of the benchmark
`reference`	Specifies the location or unique identifier of the benchmark
`QualityRatingType`	Specifies the quality rating for one filter condition.
`queryType`	Specifies the URN of the filter condition (e.g., urn:mpeg:mpqf:cs:ServiceCapabilityCS:2008:100.3.6.1 for QueryByMedia)
`rating`	Specifies the rating for the filter condition. The rating is calculated as the percentage of successful executions of queries that belong to the given filter condition. Note an exact formula depends on the used benchmark system and should be specified there.

6.10.4 Example

In the following example, the usage of `BenchmarkCapabilityType` is demonstrated within the element ProviderInformation.

```
<SchemaManagement jpsearchID="JPS-MSG-04"
                  xmlns="JPSearch:schema:management">
   <Request>
      <Output>
         <ResultItem>
            <Identifier>urn:DC-Based-Schema:1</Identifier>
            <Location>urn:DC-Based-Schema:registry:1</Location>
            <ProviderInformation>
               <Name>Schema Creator</Name>
               <Description>Image schema Developer based on MPEG
               </Description>
               <Contact>
                  <EMail>creator-email@schema.com</EMail>
                  <PostalAddress> 2 1st St., Los Angeles, CA 876554, USA
                   </PostalAddress>
                  <Phone>212-1234-4567</Phone>
                  <lang>eng</lang>
               </Contact>
               <ProviderCapability xsi:type="QueryCapabilityType">
                 <ServiceCapability>
                   <jpqf:SupportedQueryTypes href="urn:X:Y:Z"/>
                 </ServiceCapability>
               </ProviderCapability>
```

```
            <ProviderCapability xsi:type="BenchmarkCapabilityType"
                                name="Example Benchmark"
                                reference="urn:unique_identifier">
                <QualityRating queryType="urn:X:Y:Z" rating="0.3"/>
            </ProviderCapability>
          </ProviderInformation>
        </ResultItem>
      </Output>
   </Request>
</SchemaManagement>
```

6.11 ExtensionCapabilityType

6.11.1 Introduction

The `ExtensionCapabilityType` type extends the `ProviderCapabilityType` and provides a container for any description based on a specific schema specified by the namespace declaration within the description.

6.11.2 Syntax

```
<complexType name="ExtensionCapabilityType">
  <complexContent>
    <extension base="jps_mngt:ProviderCapabilityType">
      <sequence>
       <any namespace="##any" processContents="lax" minOccurs="0"
            maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

6.11.3 Semantics

Semantics of the `ExtensionCapabilityType` type:

Name	*Definition*
`ExtensionCapabilityType`	Specifies the extensibility pattern for domain or provider specific technical information. The type is a container for any description based on a specific schema specified by the namespace declaration within the description.

6.11.4 Example

In the following example, the usage of `ExtensionCapabilityType` is demonstrated within the element ProviderInformation. In this example, the provider integrated capability descriptions about its network connections.

```
<SchemaManagement jpsearchID="JPS-MSG-04"
                  xmlns="JPSearch:schema:management">
   <Request>
      <Output>
```

```
          <ResultItem>
             <Identifier>urn:DC-Based-Schema:1</Identifier>
             <Location>urn:DC-Based-Schema:registry:1</Location>
             <ProviderInformation>
               <Name>Schema Creator</Name>
               <Description>Image schema Developer based on MPEG
               </Description>
               <Contact>
                  <EMail>creator-email@schema.com</EMail>
                  <PostalAddress> 2 1st St., Los Angeles, CA 876554, USA
                   </PostalAddress>
                  <Phone>212-1234-4567</Phone>
                  <lang>eng</lang>
               </Contact>
               <ProviderCapability xsi:type="QueryCapabilityType">
                 <ServiceCapability>
                   <jpqf:SupportedQueryTypes href="urn:X:Y:Z"/>
                 </ServiceCapability>
               </ProviderCapability>
               <ProviderCapability xsi:type="BenchmarkCapabilityType"
                                   name="Example Benchmark"
                                   reference="urn:unique_identifier">
                   <QualityRating queryType="urn:X:Y:Z" rating="0.3"/>
               </ProviderCapability>
               <ProviderCapability xsi:type="ExtensionCapabilityType">
                  <NetworkCapability>
                     <Latency>1223</Latency>
                     <Bandwidth>1234</Bandwidth>
                  </NetworkCapability>
               </ProviderCapability>
             </ProviderInformation>
          </ResultItem>
       </Output>
    </Request>
</SchemaManagement>
```

6.12 SchemaType

6.12.1 Introduction

The `SchemaType` type provides information in order to identify a certain schema.

6.12.2 Syntax

```
<complexType name="SchemaType">
    <sequence>
        <element name="SchemaInformation"
                 type="jps_mngt:SchemaInformationType"/>
        <element name="TranslationRules"
                 type="jps_mngt:TranslationRulesType"/>
    </sequence>
</complexType>
```

6.12.3 Semantics

Semantics of the `SchemaType` type:

Name	*Definition*
`SchemaType`	Specifies a `SchemaType` type which is used for describing all information in order to register a schema and its translation rules.
`SchemaInformation`	Specifies the description of a certain schema.
`TranslationRules`	Specifies the set of translation rules which are necessary for reformulating a query from the reference metadata model to the registered target metadata model.

6.12.4 Example

In the following example, the usage of `SchemaType` is demonstrated within the element Schema

```
<SchemaManagement jpsearchID="JPS-MSG-01"
                  xmlns="JPSearch:schema:management">
   <Register>
      <Input>
         <ProviderInformation>
           <Name>Schema Creator</Name>
           <Description>Image schema Developer based on
MPEG</Description>
           <Contact>
              <EMail>creator-email@schema.com</EMail>
              <PostalAddress> 21st St., Los Angeles, CA 876554, USA
              </PostalAddress>
              <Phone>212-1234-4567</Phone>
              <lang>eng</lang>
           </Contact>
              <ProviderCapability xsi:type="QueryCapabilityType">
                <ServiceCapability>
                  <jpqf:SupportedQueryTypes href="urn:X:Y:Z"/>
                </ServiceCapability>
              </ProviderCapability>
              <ProviderCapability xsi:type="BenchmarkCapabilityType"
                                  name="Example Benchmark"
                                  reference="urn:unique_identifier">
                 <QualityRating queryType="urn:X:Y:Z" rating="0.3"/>
              </ProviderCapability>
        </ProviderInformation>
        <Schema>
           <SchemaInformation>
              <Description>new version 2009</Description>
              <SpecificationName>DC-Based-Schema</SpecificationName>
              <Version>1.0</Version>
              <Identifier>urn:DC-Based-Schema:1</Identifier>
              <Location>urn:DC-Based-Schema:registry:1</Location>
           </SchemaInformation>
           <TranslationRules>
              <TranslationRule>
                  <!-- example of one-to-one translation -->
```

```
                    <FromField>DCB-Field 1</FromField>
                    <ToField>Core-Field K</ToField>
                </TranslationRule>
                <fromFormat>urn:DC-Based-Schema:1</fromFormat>
                <toFormat>urn:JPSearchCore:1</toFormat>
            </TranslationRules>
        </Schema>
    </Input>
  </Register>
</SchemaManagement>
```

6.13 ReplaceInputType

6.13.1 Introduction

The ReplaceInputType type describes the data for an input request during a replacement operation. The requester has the opportunity to replace the stored provider information as well as the set of translation rules of a certain schema which is identified by its schema ID.

6.13.2 Syntax

```
<complexType name="ReplaceInputType">
  <sequence>
    <element name="SchemaID" type="anyURI"/>
    <choice>
      <element name="ProviderInformation"
        type="jps_mngt:ProviderInformationType"/>
      <element name="TranslationRules"
        type="jps_mngt:TranslationRulesType"/>
    </choice>
  </sequence>
</complexType>
```

6.13.3 Semantics

Semantics of the `ReplaceInputType` type:

Name	*Definition*
`ReplaceInputType`	Specifies the `ReplaceInputType` type which describes the data considered for replacement.
`SchemaID`	Specifies the schema ID (URI) which is the target for the replacement process.
`ProviderInformation`	Specifies the provider information of a schema which should be replaced by a new version.
`TranslationRules`	Specifies the translation rules for a certain schema which should be replaced by a new version.

6.13.4 Example

The following example shows an input scenario for replace operation that changes a translation rule.

```
<SchemaManagement jpsearchID="JPS-MSG-05"
            xmlns="JPSearch:schema:management">
   <Replace>
      <Input>
         <SchemaID>urn:DC-Based-Schema:1</SchemaID>
         <TranslationRules>
              <TranslationRule>
                  <!-- example of one-to-one translation -->
                  <FromField>DCB-Field 1</FromField>
                  <ToField>Core-Field K+3</ToField>
              </TranslationRule>
              <fromFormat>urn:DC-Based-Schema:1</fromFormat>
              <toFormat>urn:JPSearchCore:1</toFormat>
           </TranslationRules>
      </Input>
   </Replace>
</SchemaManagement>
```

6.14 ReplaceOutputType

6.14.1 Introduction

The ReplaceOutputType type describes the data which is transmitted in case of a response to a replacement input request.

6.14.2 Syntax

```
<complexType name="ReplaceOutputType">
  <sequence>
   <element name="Identifier" type="anyURI" minOccurs="0"/>
   <choice>
     <element name="Status" type="jps_mngt:InformationType"
        maxOccurs="unbounded"/>
     <element name="Warning" type="jps_mngt:InformationType"
        maxOccurs="unbounded"/>
     <element name="Exception" type="jps_mngt:InformationType"
        maxOccurs="unbounded"/>
   </choice>
  </sequence>
</complexType>
```

6.14.3 Semantics

Semantics of the `ReplaceOutputType` type:

Name	*Definition*
`ReplaceOutputType`	Specifies the `ReplaceOutputType` type which describes a set of messages related to the responder.
`Identifier`	Specifies the schema ID (URI) which is the target for the replacement process.
`Status`	Describes the status of the responder.
`Warning`	Describes the warning messages from the responder.
`Exception`	Describes the exceptions the responder encountered during the process.

6.14.4 Example

The following message shows an example for `ReplaceOutputType` in the element Output.

```
<SchemaManagement jpsearchID="JPS-MSG-06"
                  xmlns="JPSearch:schema:management">
   <Replace>
      <Output>
         <Identifier>urn:DC-Based-Schema:1</Identifier>
         <Status>
            <Code>status-code-y</Code>
            <Description>new trans. Rule registered and being evaluated
            </Description>
         </Status>
      </output>
   </Replace>
</SchemaManagement>
```

6.15 SchemaInformationType

6.15.1 Introduction

The `SchemaInformationType` type provides information to identify the schema and its different versions.

6.15.2 Syntax

```
<complexType name="SchemaInformationType">
  <sequence>
    <element name="Description" type="string" minOccurs="0" />
    <element name="SpecificationName" type="string" minOccurs="0" />
    <element name="Version" type="string" minOccurs="0" />
    <element name="Identifier" type="anyURI" minOccurs="0" />
    <element name="Location" type="anyURI" minOccurs="0" />
  </sequence>
</complexType>
```

6.15.3 Semantics

Semantics of the `SchemaInformationType` type:

Name	*Definition*
`SchemaInformationType`	Specifies a `SchemaInformationType` type which is used for identifying and describing a certain schema.
`Description`	Specifies the description of the schema.
`Version`	Specifies the version of the schema.
`Identifier`	Specifies a unique identifier of the schema.
`Location`	Specifies the location where the schema can be retrieved.
`SpecificationName`	Specifies the name of a standardization body and number/identifier

6.15.4 Example

The following message shows an example for `SchemaInformationType` in the element SchemaInformation.

```
<SchemaManagement jpsearchID="JPS-MSG-01"
                  xmlns="JPSearch:schema:management">
   <Register>
      <Input>
         <ProviderInformation>
           <Name>Schema Creator</Name>
           <Description>Image schema Developer based on
MPEG</Description>
           <Contact>
              <EMail>creator-email@schema.com</EMail>
              <PostalAddress> 2 1st St., Los Angeles, CA 876554, USA
              </PostalAddress>
              <Phone>212-1234-4567</Phone>
              <lang>eng</lang>
           </Contact>
              <ProviderCapability xsi:type="QueryCapabilityType">
                <ServiceCapability>
                  <jpqf:SupportedQueryTypes href="urn:X:Y:Z"/>
                </ServiceCapability>
              </ProviderCapability>
              <ProviderCapability xsi:type="BenchmarkCapabilityType"
                                  name="Example Benchmark"
                                  reference="urn:unique_identifier">
                  <QualityRating queryType="urn:X:Y:Z" rating="0.3"/>
              </ProviderCapability>
       </ProviderInformation>
       <Schema>
          <SchemaInformation>
             <Description>new version 2009</Description>
             <SpecificationName>DC-Based-Schema</SpecificationName>
```