
Information technology — JPSearch —
Part 6:
Reference software

Technologies de l'information — JPSearch —
Partie 6: Logiciel de référence

iTeh STANDARD PREVIEW
(standards.iteh.ai)

<https://standards.iteh.ai/catalog/standards/sist/24a20f78-b769-4236-b43f-bcff1732db73/iso-iec-24800-6-2012>

iTeh STANDARD PREVIEW (standards.iteh.ai)

[ISO/IEC 24800-6:2012](https://standards.iteh.ai/catalog/standards/sist/24a20f78-b769-4236-b43f-bcfl1732db73/iso-iec-24800-6-2012)

<https://standards.iteh.ai/catalog/standards/sist/24a20f78-b769-4236-b43f-bcfl1732db73/iso-iec-24800-6-2012>



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2012

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword	iv
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Overview and conventions	2
4.1 Organization of the document	2
4.2 Overview of the architecture of the ISO/IEC 24800 reference software	2
5 Reference software for the ISO/IEC 24800	3
5.1 Metadata Translation Module	3
5.2 JPQF Query Processor Module	11
5.3 Embedded Metadata Codec Module	20
5.4 Repository Import/Export Module	26
Bibliography	34

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[ISO/IEC 24800-6:2012](https://standards.iteh.ai/catalog/standards/sist/24a20f78-b769-4236-b43f-bcfl1732db73/iso-iec-24800-6-2012)

<https://standards.iteh.ai/catalog/standards/sist/24a20f78-b769-4236-b43f-bcfl1732db73/iso-iec-24800-6-2012>

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 24800-6 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* [ISO/IEC 24800-6:2012](https://standards.iteh.ai/catalog/standards/sist/24a20f78-b769-4236-b43f-bc11752db73/iso-iec-24800-6-2012)
- *Part 2: Registration, identification and management of schema and ontology* <https://standards.iteh.ai/catalog/standards/sist/24a20f78-b769-4236-b43f-bc11752db73/iso-iec-24800-6-2012>
- *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*

Information technology — JPSearch —

Part 6: Reference software

1 Scope

This part of ISO/IEC 24800 describes reference software for the normative clauses as well as utility software demonstrating the usage scenarios of ISO/IEC 24800-2 to ISO/IEC 24800-5. The information provided is applicable for determining the reference software modules available for ISO/IEC 24800-2 to ISO/IEC 24800-5 and understanding their functionality and usage. A software module provided in this part of ISO/IEC 24800 can be used either as standalone software or as a part of larger integrated software depending on the module.

2 Normative references

The following referenced documents, in whole or in part, 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/corrigenda) applies.

ISO/IEC 24800-2, *Information technology — JPSearch — Part 2: Registration, identification and management of schema and ontology*

ISO/IEC 24800-3, *Information technology — JPSearch — Part 3: Query format*

ISO/IEC 24800-4, *Information technology — JPSearch — Part 4: File format for metadata embedded in image data (JPEG and JPEG 2000)*

ISO/IEC 24800-5, *Information technology — JPSearch — Part 5: Data interchange format between image repositories*

3 Terms and definitions

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

3.1

module

software component implementing **reference software** or **utility software**

3.2

reference software

one or more **modules** utilizing normative parts of ISO/IEC 24800-2, -3, -4, -5

3.3

utility software

one or more **modules** utilizing informative parts of ISO/IEC 24800-1, -2, -3, -4, -5 and/or the usage of **reference software** within real-world applications

4 Overview and conventions

4.1 Organization of the document

In the remainder of this document, each reference and utility software module is described following the convention as below:

Module name	Name of the ZIP file with the following structure: /<directory>/<module_name>-<implementation>-<version>.zip <directory>: directory name in which the module can be found <module_name>: name of the module, e.g., Parser, Validator, etc. <implementation>: letter A, B, C, etc. for different implementations. <version>: version number, i.e., n_n_n n_n n
Functional Description	Describes the functionality the module provides.
Installation Guideline	Describes how the module can be installed/prepared to be used.
Interface Description	Describes necessary interface (e.g. command line instructions with parameters) to use the module. (standards.iteh.ai)
INPUT	Describes the input of the module. https://standards.iteh.ai/catalog/standards/sist/24a20f78-b769-4236-b43f-6c1f7520b75f/iso-iec-24800-6-2012
OUTPUT	Describes the output of the module. https://standards.iteh.ai/catalog/standards/sist/24a20f78-b769-4236-b43f-6c1f7520b75f/iso-iec-24800-6-2012
Programming Language(s)	Lists the programming language(s) in which the module is written.
Platform(s)	Lists the platforms the module has been tested on and is supposed to run on.
Dependencies	Lists the required libraries and code with version information.
Details	Lists any implementation details, such as architecture diagrams and data flows.

4.2 Overview of the architecture of the ISO/IEC 24800 reference software

The Part 2 module will allow 1) registering external metadata schemas and 2) validating if certain metadata instance is valid according to the Core Schema in combination with the registered external schemas. It can be used in combination with Part 4 and Part 5 modules to validate the metadata ingested into a JPSearch compliant system.

The Part 3 module will process JPQF queries against the repository. It can be used alone, acting over an independent image repository, or it can be used in combination with the other modules, acting over the repository generated by the usage of the other modules.

The Part 4 and Part 5 modules are responsible for the interchange of content and metadata between the JPSearch system and the external applications. The Part 4 module will allow extracting/annotating metadata

embedded within an image file. The Part 5 module will allow interchanging metadata from the whole repository or parts of it in XML or in binary format.

For each software module mentioned in the table below, next clauses describe a brief architecture of the module with description, functionality, instructions for installation and utilization, description of using provided interface which may be command line and/or graphical user interface, and examples of inputs and outputs with any additional information required to properly use the software module.

module name	type	description
Metadata Translation Module	normative	Schema registry and metadata validation (Part 2)
JPQF Query Processor Module	normative	JPQF query validation and processing (Part 3)
Embedded Metadata Codec Module	normative	Metadata embedded in image files. Annotation and extraction (Part 4)
Repository Import/Export Module	normative	Repository interchange in binary and XML formats (Part 5)

iTeh STANDARD PREVIEW

5 Reference software for the ISO/IEC 24800 (standard@iteh.ai)

5.1 Metadata Translation Module [ISO/IEC 24800-6:2012](https://standards.iso.org/iso/24800-6:2012)

5.1.1 Summary <https://standards.iteh.ai/catalog/standards/sist/24a20f78-b769-4236-b43f-bcfl732db73/iso-iec-24800-6-2012>

The JPSearch translation rules framework supports understanding XML instance documents of metadata formats by providing translation rules from an XML instance documents of metadata formats into the JPSearch core metadata format and vice versa. For this purpose, the framework bases on the following external technologies: a XML database called BaseX for managing the individual translation rules and JDOM which is an open-source library for java optimized XML data manipulation routines.

Module name	Metadata Translation Module
Functional Description	<p>The Java based translation module provides means for translating metadata descriptions of XML based metadata formats to and from JPSearch Core descriptions. Currently supported metadata formats are MPEG-7, Dublin Core and JPSearch Core. The current framework can be further extended by adding respective translation rules (XML instance documents) from and to JPSearch Core. Furthermore, the implementation has to be extended by a new TranslationToXXX module.</p> <p>Besides, the current version of the translation module supports the management (upload, download, delete) of translation rules. This is implemented by a JavaFX based GUI application which can be used as web application or standalone application.</p>
Installation Guideline	Runs as Java (web) application or can be used as Java library in projects. Detailed information is provided below.
Interface Description	
INPUT	JPQF query containing XML based metadata. The metadata has to be stored in a QueryByDescription query type. Currently MPEG-7, Dublin Core and JPSearch

	Core is supported.
OUTPUT	JPQF query containing the translated metadata format in the respective QueryByDescription query type. Translation means MPEG-7 or Dublin Core metadata are translated into JPSearch Core and JPSearch Core is either translated into MPEG-7 or Dublin Core.
Programming Language(s)	Java 1.6 and higher
Platform(s)	Any platform is supported that is aware of Java
Dependencies	JavaFX library for the visualization and Tomcat 6 or higher web server in case of using the translation framework as web application. A BaseX database for storing registered translation rules.
Details	

5.1.2 Functionality

5.1.2.1 Introduction

This Subclause describes the complete architecture of the metadata translation module. Figure 1 presents the individual components of the framework and its interplay. In the following, the components are described in more detail.

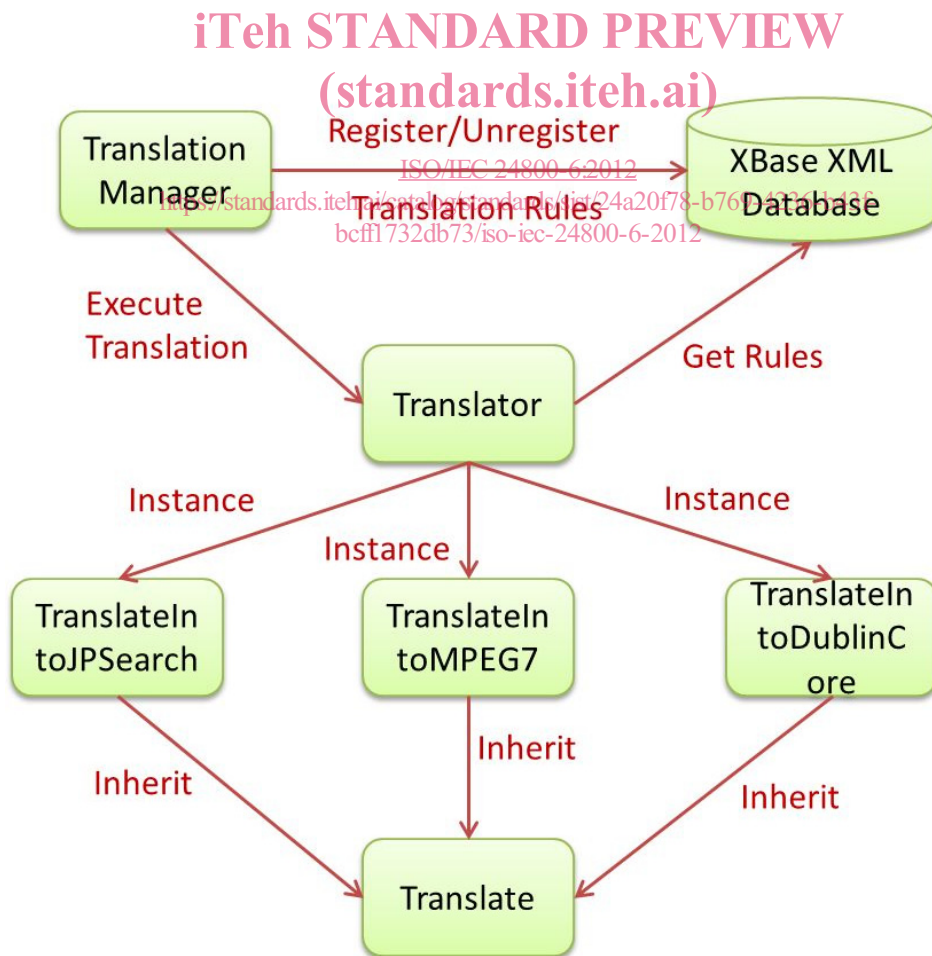


Figure 1 — Translation Rule Framework Class Hierarchy

5.1.2.2 Translation manager

The translation manager is a Java FX application which deals as frontend for user input and supports the registration of translation rules and the translation of XML input instances. Figure 2 presents a screenshot of the main frontend with the provided functionality: upload a new translation rule, view the existing rules and translate an input document.

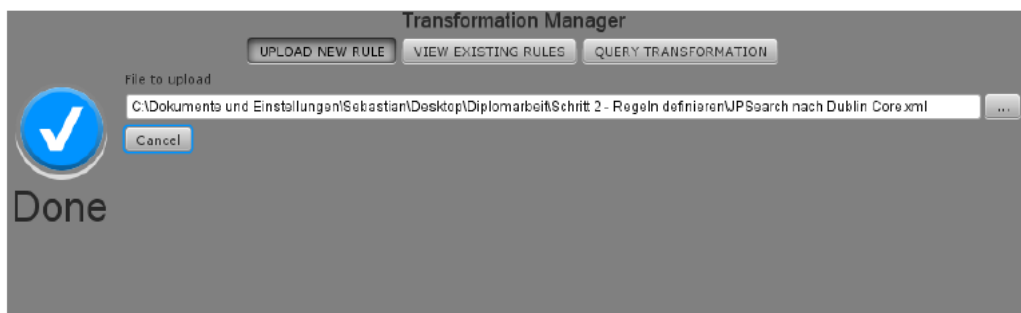


Figure 2 — Translation Manager

5.1.2.3 Module Translator

In our demo application the Translator class is instantiated directly by the translation manager. However, it can be used and instantiated in any other Java based application as well. As input parameters, the class requires a JPQF query and one or more namespaces denoting the target metadata format. Several constructors are available for different settings. If only a query is transmitted, the translator assumes to translate the content of the input query into the JPSearch core metadata format. In this case, the responsible class (*TranslateIntoJPSearch*) is used. The needed translation rules are (if not submitted by parameters) loaded from the BaseX database.

5.1.2.4 Module TranslateIntoXXX

For every supported metadata format a specific Java class has to be provided. This class follows a name convention like *TranslateTo<MetadataFormat>*. For instance for a translation into the Dublin Core schema a respective *TranslateIntoDublinCore* need to be implemented. The main task of those classes is to construct a valid XML instance document regarding to the target scheme (here for instance Dublin Core) and the given translation rule.

5.1.2.5 Module Translate

The module Translate contains the whole application logic for applying the translation rules. The class *Translate* receives the input query and the translation rules. After preparation of the query, the *executeTranslationRules* method is executed. This method traverses over all translation rules and calls the fitting rule paradigm (one-to-one, one-to-many, or many-to-one). In general, the available XPath expression is evaluated by using the JDOM-API in order to separate the desired value. The processing of the fromField expression is done automatic but for the evaluation of the *toField* the respective helper classes (*TranslateInto<MetadataFormat>*) are required. In addition, it has to be noted that the order of the translation rules is important for the final validity of the received instance document.

5.1.2.6 Integration of new metadata formats

For the integration of new metadata formats, a respective *TranslateIntoXXX* implementation has to be provided. In addition, the necessary translation rules for the new metadata formats need to be established

containing all XPath rules for mapping a JPSearch core schema instance document into the target schema and vice versa. Finally, the *TranslateIntoXXX* implementation has to be integrated into the project and the rules file can be uploaded by the user interface.

5.1.3 Installation / Utilization

The JPSearch Metadata Translation Module comes as a set of zip file and its implementation relies on a Java 1.6 installation on the target computer. Furthermore, for using the Java FX client application as a web application a running Apache Tomcat is necessary. Furthermore, a BaseX database has to be installed in case the web application is used.

The package contains the following zip files

1. Rules.zip: Contains the four necessary translation rule XML files.
2. Schema.zip: contains the according XML scheme files for MPEG-7, Dublin Core and JPSearch.
3. Querys.zip: contains 3 example queries for testing
4. Applet.zip: contains the complete Java project
5. Translation Manager.zip: contains the Java project for the JavaFx user interface
6. WebApp.zip: contains the .war file which can be directly exported to a Tomcat server

The following steps have to be applied in order to start the web project:

1. copy the transformer.war into the webapps folder of the Tomcat installation (assuming the Tomcat is already running and automatic deployment of web application is activated)
2. start the application by using your favorite browser with the following URL: <http://localhost:8080/transformer>
3. The Upload, Download and Delete Button are self explanatory.
4. For translating a query, one can select a given one from the example set and the name space field has to be selected as follows:
 - a. urn:mpeg:mpeg7:schema:2004 in case a JPSearch Query to MPEG-7 should be translated.
 - b. http://purl.org/dc/elements/1.1/ in case a JPSearch Query to Dublin Core should be translated.
 - c. Keep the field empty in case any Query should be translated into JPSearch core schema.

For using this module as API, one of the constructors of the Translator.class file should be used as starter.

5.1.4 Metadata Comparison

This Subclause provides information about the implemented translation guideline of the supported metadata formats, namely MPEG-7 and Dublin Core. Here, the identified corresponding elements of the JPSearch core schema and the respective target schema are modeled. In detail, the MPEG-7 schema and Dublin Core has been used. For each metadata format, the semantic relations along with the mappings are specified.

Table 1 — Semantic relations of JPSearch elements and MPEG-7 elements

JPSearch element	core	Semantic relation	MPEG-7 XPath
Identifier		more generic	//DescriptionMetadata/PublicIdentifier OR //MediaInformation/MediaIdentification /EntityIdentifier
Modifiers		more specific	//CreationInformation/Creation/Creator/Role [@href=""modifier""]/following-sibling::Agent [@type=""PersonType""] /Name

Creators	exact	//CreationInformation/Creation/Creator/Role [@href="creator"]/following-sibling::Agent [@type="PersonType"]/Name
Publisher	more specific	//CreationInformation/Creation/Creator/Role [@href="publisher"]/following-sibling::Agent OR //UsageInformation/Availability/Dissemination /Disseminator/Role[@href="publisher"]/Agent
CreationDate	exact	//CreationInformation/Creation/ CreationCoordinates/Date/TimePoint
ModifiedDate	more generic	//DescriptionMetadata/LastUpdate
Description	more generic	//CreationInformation/Creation/Abstract /FreeTextAnnotation
RightsDescription	more specific	//CreationInformation/Creation/CopyrightString OR //UsageInformation/Rights
Source	more specific	//Variation/Source OR //MediaInformation/MediaIdentification /EntityIdentifier
Keyword	exact	//CreationInformation/Classification/Subject/ KeywordAnnotation/Keyword
Title	exact	//CreationInformation/Creation/Title
CollectionLabel	exact	//DescriptionUnit[@type="DescriptorCollectionType"] /@name
PreferenceValue	more generic	//UserPreferences/FilteringAndSearchPreferences /CreationPreferences/attribute::preferenceValue
Rating	exact	//CreationInformation/Classification/MediaReview /Rating
OriginalImage	more specific	//Variation/Source*/MediaLocator/MediaUri
GPSPositioning	more specific	//Semantics/SemanticBase[@type="SemanticPlaceType"] /Place/GeographicPosition/Point/
RegionOfInterest	more specific	//SpatialDecomposition/StillRegion AND //Semantics/SemanticBase

Table 2 — Corresponding elements of JPSearch elements and MPEG-7 elements

JPSearch element	core	JPSearch element	MPEG-7 element
Identifier		Identifier (anyURI)	PublicIdentifier (anyURI) OR EntityIdentifier (anyURI)
Modifiers		GivenName (string) FamilyName (string)	GivenName (string) FamilyName (string)
Creators		GivenName (string) FamilyName (string)	GivenName (string) FamilyName (string)
Publisher		PersonName/GivenName (string)	Name/GivenName (string)
		PersonName/FamilyName (string)	Name/FamilyName (string)
		OrganizationInformation/Name (string)	Name (string)
		OrganizationInformation/Addresses/Name (string)	Address/Name (string)
		OrganizationInformation/Addresses/Description (string) &	Address/PlaceDescription (string)
CreationDate		CreationDate (dateTime)	TimePoint (dateTime)
ModifiedDate		ModifiedDate (dateTime)	LastUpdate (dateTime)
Description		Description (string)	FreeTextAnnotation (string)
RightsDescription		Description (string)	RightsID (string)
Source		SourceElementType (string)	
		SourceElement/SourceElement Title (string)	
		SourceElement/SourceElement-Description (string)	
		SourceElement/SourceElement Identifier (anyURI)	*/MediaLocator/MediaUri (anyURI)
Keyword		Keyword (string)	Keyword (string)
Title		Title (string)	Title (string)
CollectionLabel		CollectionLabel (string)	attribute::name (string)
PreferenceValue		PreferenceValue (integer)	attribute::preferenceValue (integer)
Rating		LabelDefinition (anyURI)	
		LabelValue (string)	RatingValue (float)
OriginalImage		OriginationOfID (anyURI)	MediaUri (anyURI)
		Identifier	
GPSPositioning		attribute::longitude (double)	attribute::longitude (double)
		attribute::latitude (double)	attribute::latitude (double)
		attribute::altitude (double)	attribute::altitude (double)
RegionOfInterest		RegionLocator/Region (IntegerMatrixType)	StillRegion/SpatialLocator/Box (integer list)

Description (string)	StillRegion/CreationInformation /Creation/Abstract/FreeTextAnnotation (string)
Keyword (string)	StillRegion/CreationInformation /Classification/Subject/KeywordAnnotation/Keyword (string)
Title (string)	StillRegion/CreationInformation /Creation/Title (string)
ContentDescription/Person/Name/GivenName (string)	SemanticBase["@xsi:type="mpeg7:AgentObjectType"]/Agent /Name/GivenName (string)
ContentDescription/Person/Name/FamilyName (string)	SemanticBase["@xsi:type="mpeg7:AgentObjectType"]/Agent /Name/FamilyName (string)
ContentDescription/Person/Affiliation/Name (string)	SemanticBase["@xsi:type="mpeg7:AgentObjectType"]/Agent["@xsi:type="mpeg7:OrganizationType"]/Name (string)
ContentDescription/Person/Affiliation/Address/Name (string)	SemanticBase["@xsi:type="mpeg7:AgentObjectType"]/Agent["@xsi:type="mpeg7:OrganizationType"]/Address/Name (string)
ContentDescription/Person/Affiliation/Address/Description (string)	SemanticBase["@xsi:type="mpeg7:AgentObjectType"]/Agent["@xsi:type="mpeg7:OrganizationType"]/Address/PlaceDescription (string)
ContentDescription/Person/Address/Name (string)	SemanticBase["@xsi:type="mpeg7:AgentObjectType"]/Agent["@xsi:type="mpeg7:PersonType"]/Address/Name (string)
ContentDescription/Person/Address/Description (string) &	SemanticBase["@xsi:type="mpeg7:AgentObjectType"]/Agent["@xsi:type="mpeg7:PersonType"]/Address/PlaceDescription (string)
ContentDescription/Person/Description (string)	SemanticBase["@xsi:type="mpeg7:AgentObjectType"]/Agent["@xsi:type="mpeg7:PersonType"]/PersonDescription (string)
ContentDescription/Person/Nationality (string)	SemanticBase["@xsi:type="mpeg7:AgentObjectType"]/Agent["@xsi:type="mpeg7:PersonType"] /Nationality ([a-zA-Z])
ContentDescription/Object/Name (string) &	SemanticBase["@xsi:type="mpeg7:ObjectType"]/Object/Label/Name (string)
ContentDescription/Place/Name (string)	SemanticBase["@xsi:type="mpeg7:SemanticPlaceType"]/Place/Name (string)
ContentDescription/Place/Description (string)	SemanticBase["@xsi:type="mpeg7:SemanticPlaceType"]/Place/PlaceDescription (string)
ContentDescription/Event/Label (anyURI)	SemanticBase["@xsi:type="mpeg7:EventType"]/Event/Label/Name (string)