INTERNATIONAL STANDARD

ISO/IEC 24800-6

First edition
2012-05-01

Information technology — JPSearch —

Part 6: Reference software

Technologies de l'information — JPSearch —

Partie 6: Logiciel de référence

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

Reference number
ISO/IEC 24800-6:2012(E)

© ISO/IEC 2012

IECNORM.COM : Click to view the full PDF of 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

© ISO/IEC 2012 – All rights reserved

Contents

Page

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

© ISO/IEC 2012 – All rights reserved

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*

— *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*

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

© ISO/IEC 2012 – All rights reserved

Information technology — JPSearch —

Part 6:
Reference software

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

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 **module**s utilizing normative parts of ISO/IEC 24800-2, -3, -4, -5

3.3
utility software
one or more **module**s utilizing informative parts of ISO/IEC 24800-1, -2, -3, -4, -5 and/or the usage of **reference software** within real-world applications

© ISO/IEC 2012 – All rights reserved

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.
INPUT	Describes the input of the module.
OUTPUT	Describes the output of the module.
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.

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

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

© ISO/IEC 2012 – All rights reserved

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)

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

5 Reference software for the ISO/IEC 24800

5.1 Metadata Translation Module

5.1.1 Summary

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	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. 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.
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

© ISO/IEC 2012 – All rights reserved

	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.

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

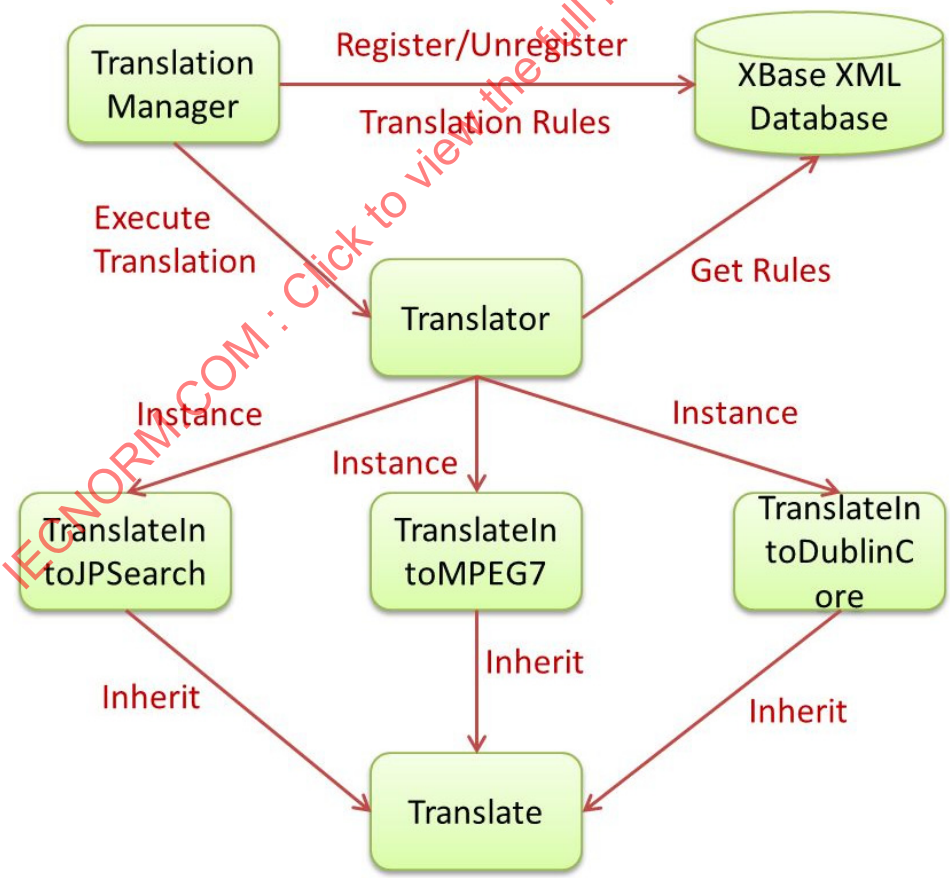

Figure 1 — Translation Rule Framework Class Hierarchy

© ISO/IEC 2012 – All rights reserved

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.

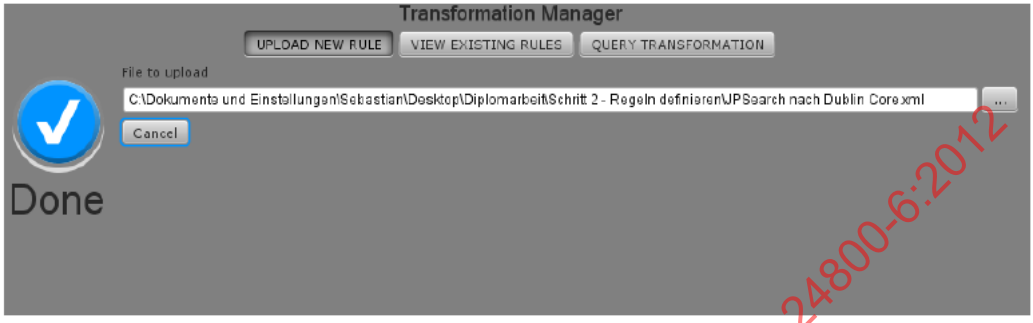

Figure 2 — Translation Manager

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

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

© ISO/IEC 2012 – All rights reserved

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 core element	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

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

© ISO/IEC 2012 – All rights reserved

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

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

© ISO/IEC 2012 – All rights reserved

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

JPSearch core element	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/Address /Name (string)	Address/Name (string)
	OrganizationInformation/Address /Description (string) &	Address/PlaceDescription (string)
CreationDate	CreationDate (dateTime)	TimePoint (dateTime)
ModifiedDate	ModfiedDate (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)

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

© ISO/IEC 2012 – All rights reserved

	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)

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

© ISO/IEC 2012 – All rights reserved

Table 3 — Semantic relations of JPSearch core schema and Dublin Core

JPSearch core element	Semantic relation	Dublin Core element
Identifier	exact	Identifier (string)
Modifiers	more generic	Contributor (string)
Creators	more generic	creator (string)
Publisher	more generic	publisher (string)
CreationDate	more generic	date (string)
ModifiedDate	more generic	date (string)
Description	exact	description (string)
RightsDescription	exact	rights (string)
Source	exact	source (string)
Keyword	exact	subject (string)
Title	exact	title (string)
CollectionLabel	not available	not available
PreferenceValue	not available	not available
Rating	not available	not available
OriginalImage	exact	relation (string)
GPSPositioning	more generic	coverage (string)
RegionOfInterest	not available	not available

Table 4 — Corresponding elements in JPSearch core schema and Dublin Core

JPSearch core element	JPSearch element	Dublin Core element
Identifier	Identifier (anyURI)	Identifier (string)
Modifiers	GivenName (string) FamilyName (string)	Contributor (string)
Creators	GivenName (string) FamilyName (string)	creator (string)
Publisher	GivenName (string) FamilyName (string)	publisher (string)
CreationDate	CreationDate (dateTime)	date (string)

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

© ISO/IEC 2012 – All rights reserved

ModifiedDate	ModfiedDate (dateTime)	date (string)
Description	Description (string)	description (string)
RightsDescription	Description (string)	rights (string)
Source	SourceElementType (string)	source (string)
Keyword	Keyword (string)	subject (string)
Title	Title (string)	title (string)
CollectionLabel	not available	not available
PreferenceValue	not available	not available
Rating	not available	not available
OriginalImage	OriginationOfID (anyURI)	relation (string)
GPSPositioning	attribute::longitude (double) attribute::latitude (double)	coverage (string)
RegionOfInterest	not available	not available

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

5.2 JPQF Query Processor Module

5.2.1 Summary

Module name	JPQF QueryProcessor
Functional Description	Checks the syntactic and semantic validity of a JPQF input/output query according to the rules of XML 1.1 and the JPQF XML schema. Interprets the basic conditions of an JPQF input query (without QueryTypes) over an example metadata file (e.g. an MPEG-7 collection of image descriptors) and returns a coherent JPQF response
Installation Guideline	Copy the Java jar file to the local filesystem. Requires a previous Java 1.6 installation and the operating system execution path environment variable properly pointing to the Java application launcher (java command).
Interface Description	*java -Dlog4j.configuration=file:./WEB-INF/classes/log4j.properties -classpath ./WEB-INF/lib/mpqf-1.0.jar;./WEB-INF/lib/xmldb.jar;./WEB-INF/lib/exist.jar;./WEB-INF/lib/log4j-1.2.15.jar;./WEB-INF/lib/xmlrpc-1.2-patched.jar;./WEB-INF/lib/jaxen-1.1.1.jar;./WEB-INF/lib/commons-pool-1.4.jar;./WEB-INF/lib/antlr-2.7.6.jar;./WEB-INF/lib/xercesImpl-2.9.1.jar;./WEB-INF/lib/resolver-1.2.jar;./WEB-INF/lib/quartz-1.6.0.jar;./WEB-INF/lib/commons-logging-1.0.4.jar;./WEB-INF/lib/jta.jar;./WEB-INF/lib/commons-collections-3.1.jar;./WEB-INF/lib/stax-api-1.0.1.jar;./WEB-INF/lib/caliph-emir-cbir.jar;./WEB-INF/lib/lucene-core-2.1.0.jar;./WEB-INF/lib/lire.jar* ***org.barcelonatech.kaikoMPQFTester [test_jpsearch_p3_query.xml] [test_jpsearch_p5_interchangefile.xml] [outputfile.xml]***

© ISO/IEC 2012 – All rights reserved

INPUT	A JPQF query; An example metadata file
OUTPUT	A JPQF response
Programming Language(s)	Java 1.6
Platform(s)	Windows, Linux and any other platform supporting Java 1.6.
Dependencies	NO
Details	Module based in the reference software module described in ISO/IEC 15938-12:2008/Amd.1, which provides an MPEG Query Format Basic Interpreter. The provided JPQF Query Processor module extends the software in ISO/IEC 15938-12:2008/Amd.1 by adjusting the parameters of the query processor to allow the compilation and execution of queries with features which are specific of ISO/IEC 24800-3, and also by allowing the execution of the queries over metadata compliant with ISO/IEC 24800-2.

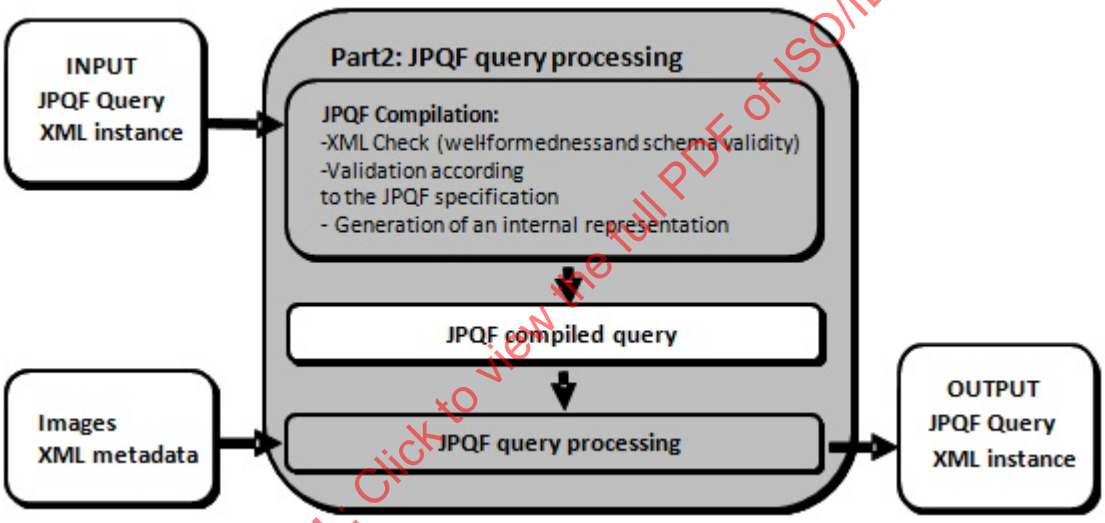

Figure 3 — JPQF query validation and processing module architecture overview

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

5.2.2 Functionality

The provided JPQF Query Processor module is based in the reference software module described in ISO/IEC 15938-12:2008/Amd.1, which provides an MPEG Query Format Basic Interpreter. The provided JPQF Query Processor module extends the software in ISO/IEC 15938-12:2008/Amd.1 by adjusting the parameters of the query processor to allow the compilation and execution of queries with features which are specific of ISO/IEC 24800-3, and also by allowing the execution of the queries over metadata compliant with ISO/IEC 24800-2.

© ISO/IEC 2012 – All rights reserved

The table below lists the features which are covered by the provided software.

Feature	Description
Query compilation (parsing)	Syntactic and semantic validation of a text query. Generation of a compiled (parsed) query in the form of an object structure, traversable by a programming language.
Basic conditions	AND, OR, NOT, XOR, comparison expressions.
Granularity	Different granularities specified with the *EvaluationPath* element below the *QueryCondition* element
Sorting	Any possible usage of the *SortByFieldType* and *SortByAggregateType*
Grouping	Any possible usage of the *GroupBy* element
Join	JoinType with the evaluation*Path* element
Combination of Grouping and Join	GroupBy + Join
Compliant with Part 2	Address metadata paths according to ISO/IEC 24800-2
Compliant with Part 5	Accept input metadata formatted according to ISO/IEC 24800-5

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

5.2.3 Command line utilization

5.2.3.1 Instruction for commands

This module provides a standalone basic interpreter which allows command line testing of JPQF queries over a single ISO/IEC 24800-5 metadata file containing the description of multiple images.

The JPQF Query Processor executable comes as a Java jar file and relies on a Java 1.6 (or higher) installation on the target computer.

The standalone test version of the interpreter can be executed by the following command:

java -Dlog4j.configuration=file:./WEB-INF/classes/log4j.properties -classpath ./WEB-INF/lib/mpqf-1.0.jar;./WEB-INF/lib/xmldb.jar;./WEB-INF/lib/exist.jar;./WEB-INF/lib/log4j-1.2.15.jar;./WEB-INF/lib/xmlrpc-1.2-patched.jar;./WEB-INF/lib/jaxen-1.1.1.jar;./WEB-INF/lib/commons-pool-1.4.jar;./WEB-INF/lib/antlr-2.7.6.jar;./WEB-INF/lib/xercesImpl-2.9.1.jar;./WEB-INF/lib/resolver-1.2.jar;./WEB-INF/lib/quartz-1.6.0.jar;./WEB-INF/lib/commons-logging-1.0.4.jar;./WEB-INF/lib/jta.jar;./WEB-INF/lib/commons-collections-3.1.jar;./WEB-INF/lib/stax-api-1.0.1.jar;./WEB-INF/lib/caliph-emir-cbir.jar;./WEB-INF/lib/lucene-core-2.1.0.jar;./WEB-INF/lib/lire.jar ***org.barcelonatech.kaikoMPQFTester [test_jpsearch_p3_query.xml] [test_jpsearch_p5_interchangefile.xml] [outputfile.xml]***

A .bat/.sh script which instantiates this commands with two parameters is provided within the ***test/jpsearch_tests*** directory:

jpqf.bat ***[test_jpsearch_p3_query.xml] [test_jpsearch_p5_interchangefile.xml] [outputfile.xml]***

© ISO/IEC 2012 – All rights reserved

The directory *test/jpsearch_tests* contains several test queries and a test ISO/IEC 24800-5 file with the descriptions of different images. For example:

jpqf.bat test1_1_emptyquery.xml images_metadata_28400_5.xml

5.2.3.2 Example Utilization

jpqf.bat test2_1_comparison_equal.xml images_metadata_28400_5.xml out.xml

Example input query (*test2_1_comparison_equal.xml)*:

```
<?xml version="1.0" encoding="UTF-8"?>
<JPEGQuery xmlns="urn:jpeg:jpqf:schema:2008"
xmlns:mpqf="urn:mpeg:mpqf:schema:2008"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:jpeg:jpqf:schema:2008 24800-3.xsd
urn:mpeg:mpqf:schema:2008 mpqf_cor1_cor2_amd1.xsd"
jpqfID="http://www.mpqf.org/id1">
 <InputQuery>
    <OutputDescription maxItemCount="3" maxPageEntries="10"
outputNameSpace="urn:mpeg:mpeg7:schema:2004">
   <mpqf:ReqField typeName="jpcs:uri">../../ImageData/MediaUri</mpqf:ReqField>
   <mpqf:ReqField typeName="jpcs:identifier">Identifier</mpqf:ReqField>
   <mpqf:ReqField
typeName="jpcs:Creators/GivenName">Creators/GivenName</mpqf:ReqField>
   <mpqf:ReqField
typeName="jpcs:Creators/FamilyName">Creators/FamilyName</mpqf:ReqField>
   <mpqf:ReqField typeName="jpcs:CreationDate">CreationDate</mpqf:ReqField>
   <mpqf:ReqField typeName="latitude">GPSPositioning/@latitude</mpqf:ReqField>
   <mpqf:ReqField typeName="longitude">GPSPositioning/@longitude</mpqf:ReqField>
  </OutputDescription>
      <QueryCondition>

<EvaluationPath>/ImageRepository/Image/ImageMetadata/JPSearchImageDescription</Ev
aluationPath>
          <Condition xsi:type="mpqf:Equal">
            <mpqf:DateTimeField>CreationDate</mpqf:DateTimeField>
            <mpqf:DateTimeValue>2009-10-07T08:46:45</mpqf:DateTimeValue>
          </Condition>
      </QueryCondition>
    </InputQuery>
</JPEGQuery>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

© ISO/IEC 2012 – All rights reserved

Example ISO/IEC 24800-5 doc (*test/jpsearch_tests/images_metadata_28400_5.xml)*:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<ImageRepository xmlns="urn:jpeg:jpsearch:jpxif:2009"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:jpcs="urn:jpeg:jpsearch:schema:coremetadata:2009"
xmlns:jpcls="urn:jpeg:jpsearch:schema:collections:2009"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004"
xsi:schemaLocation="urn:jpeg:jpsearch:jpxif:2009 24800-5-repository.xsd
urn:jpeg:jpsearch:schema:coremetadata:2009 24800-2-core.xsd
urn:jpeg:jpsearch:schema:collections:2009 24800-5-collections.xsd
urn:mpeg:mpeg7:schema:2004 M7v2schema.xsd">
 <Image>
  <ImageData>
   <MediaUri>imageDB/test/imageFiles/testimage1.jpg</MediaUri>
  </ImageData>
  <ImageMetadata>
   <JPSearchImageDescription>
    <jpcs:Identifier>urn:unique:identifier:1</jpcs:Identifier>
    <jpcs:Creators xml:lang="en-us">
     <jpcs:GivenName>John</jpcs:GivenName>
     <jpcs:FamilyName>Smith</jpcs:FamilyName>
    </jpcs:Creators>
    <jpcs:CreationDate>2009-10-07T08:46:45</jpcs:CreationDate>
    <jpcs:Description>String</jpcs:Description>
    <jpcs:Keyword>String</jpcs:Keyword>
    <jpcs:Title>String</jpcs:Title>
    <jpcs:CollectionLabel>String</jpcs:CollectionLabel>
    <jpcs:PreferenceValue>2</jpcs:PreferenceValue>
    <jpcs:GPSPositioning latitude="12.0" longitude="12.0"/>
    <jpcs:RegionOfInterest>
     <jpcs:RegionLocator>
      <jpcs:Region dim="2">0 0 100 100</jpcs:Region>
     </jpcs:RegionLocator>
     <jpcs:ContentDescription/>
     <jpcs:ExternalDescription>
      <jpcs:TagName>aa</jpcs:TagName>
      <jpcs:LiteralValue>123412</jpcs:LiteralValue>
     </jpcs:ExternalDescription>
     <jpcs:ExternalDescription>
      <jpcs:TagName>aa</jpcs:TagName>
      <jpcs:StructuredValue fromNamespace="urn:mpeg:mpeg7:schema:2004">
       <mpeg7:Mpeg7>
        <mpeg7:DescriptionUnit xsi:type="mpeg7:MediaIdentificationType">
         <mpeg7:EntityIdentifier/>
        </mpeg7:DescriptionUnit>
       </mpeg7:Mpeg7>
      </jpcs:StructuredValue>
     </jpcs:ExternalDescription>
     </jpcs:RegionOfInterest>
    <jpcs:Width>640</jpcs:Width>
    <jpcs:Height>480</jpcs:Height>
   </JPSearchImageDescription>
  </ImageMetadata>
 </Image>
 <Image>
  <ImageData>
   <MediaUri>imageDB/test/imageFiles/testimage2.jpg</MediaUri>
  </ImageData>
  <ImageMetadata>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

© ISO/IEC 2012 – All rights reserved

```
   <JPSearchImageDescription>
    <jpcs:Identifier>urn:unique:identifier:2</jpcs:Identifier>
    <jpcs:Creators xml:lang="en-us">
     <jpcs:GivenName>Jack</jpcs:GivenName>
     <jpcs:FamilyName>Black</jpcs:FamilyName>
    </jpcs:Creators>
    <jpcs:CreationDate>2001-12-17T09:30:47.0Z</jpcs:CreationDate>
    <jpcs:Description>String</jpcs:Description>
    <jpcs:Keyword>String</jpcs:Keyword>
    <jpcs:Title>String</jpcs:Title>
    <jpcs:CollectionLabel>String</jpcs:CollectionLabel>
    <jpcs:PreferenceValue>1</jpcs:PreferenceValue>
    <jpcs:GPSPositioning latitude="12.0" longitude="12.0"/>
    <jpcs:RegionOfInterest>
     <jpcs:RegionLocator>
      <jpcs:Region dim="2">0 0 100 100</jpcs:Region>
     </jpcs:RegionLocator>
     <jpcs:ContentDescription/>
     <jpcs:ExternalDescription>
      <jpcs:TagName>aa</jpcs:TagName>
      <jpcs:LiteralValue>123412</jpcs:LiteralValue>
     </jpcs:ExternalDescription>
     <jpcs:ExternalDescription>
      <jpcs:TagName>aa</jpcs:TagName>
      <jpcs:StructuredValue fromNamespace="urn:mpeg:mpeg7:schema:2004">
       <mpeg7:Mpeg7>
        <mpeg7:DescriptionUnit xsi:type="mpeg7:MediaIdentificationType">
         <mpeg7:EntityIdentifier/>
        </mpeg7:DescriptionUnit>
       </mpeg7:Mpeg7>
      </jpcs:StructuredValue>
     </jpcs:ExternalDescription>
    </jpcs:RegionOfInterest>
    <jpcs:Width>640</jpcs:Width>
    <jpcs:Height>480</jpcs:Height>
   </JPSearchImageDescription>
  </ImageMetadata>
</Image>
<Image>
 <ImageData>
  <MediaUri>imageDB/test/imageFiles/testimage3.jpg</MediaUri>
 </ImageData>
 <ImageMetadata>
  <JPSearchImageDescription>
   <jpcs:Identifier>urn:unique:identifier:3</jpcs:Identifier>
   <jpcs:Creators xml:lang="en-us">
    <jpcs:GivenName>Smith</jpcs:GivenName>
    <jpcs:FamilyName>John</jpcs:FamilyName>
   </jpcs:Creators>
   <jpcs:CreationDate>2001-12-17T09:30:47.05</jpcs:CreationDate>
   <jpcs:Description>String</jpcs:Description>
   <jpcs:Keyword>String</jpcs:Keyword>
   <jpcs:Title>String</jpcs:Title>
   <jpcs:CollectionLabel>String</jpcs:CollectionLabel>
   <jpcs:PreferenceValue>1</jpcs:PreferenceValue>
   <jpcs:GPSPositioning latitude="10.0" longitude="12.0"/>
   <jpcs:Width>640</jpcs:Width>
   <jpcs:Height>480</jpcs:Height>
  </JPSearchImageDescription>
 </ImageMetadata>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

© ISO/IEC 2012 – All rights reserved

```
 </Image>
 <Image>
  <ImageData>
   <MediaUri>imageDB/test/imageFiles/testimage4.jpg</MediaUri>
  </ImageData>
  <ImageMetadata>
   <JPSearchImageDescription>
    <jpcs:Identifier>urn:unique:identifier:4</jpcs:Identifier>
    <jpcs:Creators xml:lang="en-us">
     <jpcs:GivenName>Peter</jpcs:GivenName>
     <jpcs:FamilyName>Wei</jpcs:FamilyName>
    </jpcs:Creators>
    <jpcs:CreationDate>2001-12-17T09:30:47.00</jpcs:CreationDate>
    <jpcs:Description>String</jpcs:Description>
    <jpcs:Keyword>String</jpcs:Keyword>
    <jpcs:Title>String</jpcs:Title>
    <jpcs:CollectionLabel>String</jpcs:CollectionLabel>
    <jpcs:PreferenceValue>5</jpcs:PreferenceValue>
    <jpcs:GPSPositioning latitude="10.0" longitude="12.0"/>
    <jpcs:Width>640</jpcs:Width>
    <jpcs:Height>480</jpcs:Height>
   </JPSearchImageDescription>
  </ImageMetadata>
 </Image>
 <Image>
  <ImageData>
   <MediaUri>imageDB/test/imageFiles/testimage5.jpg</MediaUri>
  </ImageData>
  <ImageMetadata>
   <JPSearchImageDescription>
    <jpcs:Identifier>urn:unique:identifier:5</jpcs:Identifier>
    <jpcs:Creators xml:lang="en-us">
     <jpcs:GivenName>Peter</jpcs:GivenName>
     <jpcs:FamilyName>Wei</jpcs:FamilyName>
    </jpcs:Creators>
    <jpcs:CreationDate>2001-12-17T09:30:47.00</jpcs:CreationDate>
    <jpcs:Description>String</jpcs:Description>
    <jpcs:Keyword>String</jpcs:Keyword>
    <jpcs:Title>String</jpcs:Title>
    <jpcs:CollectionLabel>String</jpcs:CollectionLabel>
    <jpcs:PreferenceValue>1</jpcs:PreferenceValue>
    <jpcs:GPSPositioning latitude="10.0" longitude="12.0"/>
    <jpcs:Width>640</jpcs:Width>
    <jpcs:Height>480</jpcs:Height>
   </JPSearchImageDescription>
  </ImageMetadata>
 </Image>
</ImageRepository>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

Example resulting output (*out.xml)*:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<JPEGQuery xmlns="urn:jpeg:jpqf:schema:2008" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:mpqf="urn:mpeg:mpqf:schema:2008" xsi:schemaLocation="urn:jpeg:jpqf:schema:2008 24800-3.xsd
urn:mpeg:mpqf:schema:2008 mpqf_cor1_cor2_amd1.xsd" jpqfID="http://www.mpqf.org/id1">
    <OutputResult currPage="1" totalPages="1" expirationDate="2011-07-22T00:00:00">
        <mpqf:ResultItem recordNumber="1">
            <mpqf:TextResult/>
```

© ISO/IEC 2012 – All rights reserved

```
            <mpqf:MediaResource/>
            <mpqf:FragmentResult
name="jpcs:uri">imageDB/test/imageFiles/testimage1.jpg</mpqf:FragmentResult>
            <mpqf:FragmentResult name="jpcs:identifier">urn:unique:identifier:1</mpqf:FragmentResult>
            <mpqf:FragmentResult name="jpcs:Creators/GivenName">John</mpqf:FragmentResult>
            <mpqf:FragmentResult name="jpcs:Creators/FamilyName">Smith</mpqf:FragmentResult>
            <mpqf:FragmentResult name="jpcs:CreationDate">2009-10-07T08:46:45</mpqf:FragmentResult>
            <mpqf:FragmentResult name="latitude">12.0</mpqf:FragmentResult>
            <mpqf:FragmentResult name="longitude">12.0</mpqf:FragmentResult>
        </mpqf:ResultItem>
    </OutputResult>
</JPEGQuery>
```

5.2.4 Embedding the module

5.2.4.1 Introduction

The provided query processor can be used also as embedded in another Java application. The software is divided in several packages, but only one is necessary to access the functionality of the query processor, the package org.iso.mpeg.mpqf. This package contains a set of generic public classes and interfaces which allow indexing content and metadata, and executing MPQF and JPQF Query requests. If an application uses only org.iso.mpeg.mpqf classes and interfaces, it keeps decoupled from the internal implementation of the query engine, and the metadata and content indexes. This philosophy pursues the maximum interoperability, and allows combining a query engine, a metadata index, and a content index from three different third parties. Any one of the components can be replaced with minimum impact to the host application. The package org.iso.mpeg.mpqf is bundled along with a specific implementation of a query engine, and basic metadata and content indexes from third parties.

A javadoc with the API details has been generated and included in the application bundle.

5.2.4.2 Interface Hierarchy

- org.iso.mpeg.mpqf.**MPQFEngine**
- org.iso.mpeg.mpqf.**ContentIndex**
- org.iso.mpeg.mpqf.**XMLIndex**

5.2.4.3 Class Hierarchy

- java.lang.Object
    - org.iso.mpeg.mpqf.**MPQFEngineFactory**
    - org.iso.mpeg.mpqf.**MPQFOutput**
    - org.iso.mpeg.mpqf.**MPQFQuery**
    - org.iso.mpeg.mpqf.**ResultItem**
    - java.lang.Throwable (implements java.io.Serializable)
        - java.lang.Exception
            - org.iso.mpeg.mpqf.**MPQFException**
    - org.iso.mpeg.mpqf.**ContentIndexFactory**
    - org.iso.mpeg.mpqf.**XMLIndexFactory**

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

© ISO/IEC 2012 – All rights reserved

5.2.4.4 Interface MPQFEngine

This is the main component of the API. A particular class implementing the methods of the MPQFEngine interface can be obtained from the MPQFEngineFactory. By default, the UPC – BARCELONA TECH implementation will be used. The pubic methods of MPQFEngine are:

- MPQFQuery compileQuery(java.io.File mpqfFile)
- MPQFQuery compileQuery(java.io.InputStream mpqfQueryStream)
- MPQFQuery compileQuery(java.lang.String mpqfString)
- MPQFOutput executeQuery(MPQFQuery query)
- void setXMLIndex(XMLIndex index)
- void setContentIndex(ContentIndex contentIndex)

Usage steps:

1) Obtain an implementation through the MPQFEngineFactory
2) Register an XMLIndex and a ContentIndex through the setXMLIndex and setContentIndex methods
3) Index some content and metadata (see XMLIndex and ContentIndex interfaces)
4) Compile a query through one of the compileQuery methods
5) Execute the query through the executeQuery method

5.2.4.5 Interface XMLIndex

This interface allows connecting the query engine with external XML metadata databases. A specific class implementing the XMLIndex interface can be obtained from the XMLIndexFactory. The only pubic method necessary for basic usage is:

void indexMetadata(java.io.InputStream xmlStream)

Other methods are available for those who want to implement their own query processors and need to interact with the XML DB. These methods are documented in the javadoc.

5.2.4.6 Interface ContentIndex

This interface allows connecting the query engine with external content databases (video, still images, audio etc.). A specific class implementing the ContentIndex interface can be obtained from the ContentIndexFactory. . The only pubic method necessary for basic usage is:

void indexContent(java.io.InputStream contentStream)

Other methods are available for those who want to implement their own query processors and need to interact with the content DB. These methods are documented in the javadoc.

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

© ISO/IEC 2012 – All rights reserved

5.2.5 Example Utilization

Example Java code

```
MPQFEngine mpqfEngine =
    MPQFEngineFactory.createMPQFEngine("org.barcelonatech.kaiko.MPQFEngineImplUPC
");

//Setup XMLIndex
XMLIndex xmlIndex =
    xmlIndex =
XMLIndexFactory.createXMLIndex("org.barcelonatech.kaiko.existdriver.
                XMLIndexImplExistEmbedded", "imageDB_index/blank/index-exist",
true);
//Index metadata file
InputStream xmlStream = new FileInputStream("example_mpeg7.xml"));
xmlIndex.indexMetadata(xmlStream);

//Register XMLIndex
mpqfEngine.setXMLIndex(xmlIndex);

//Query compilation
MPQFQuery mpqfQuery = mpqfEngine.compileQuery(new File(args[0]));

//Query execution
MPQFOutput mpqfOutput = mpqfEngine.executeQuery(mpqfQuery);

for (int i = 0; i<mpqfOutput.resultItemVector.size(); i++) {
    ResultItem ri = (ResultItem)mpqfOutput.resultItemVector.elementAt(i);
    System.out.println("ResultItem:");
    System.out.println(ri.textResult);
    System.out.println(ri.mediaResource);
    System.out.println();
}
```

5.3 Embedded Metadata Codec Module

5.3.1 Summary

Module name	Embedded Metadata Codec: JPSearch File Format
Functional Description	The JPSearch File Format library provides means for integrating metadata descriptions in various formats (XML based such as MPEG-7 or the JPSearch Core) into a JPEG or JPEG 2000 encoded image file which is stored as JPSearch File Format. The encoded file can be viewed and processed by any JPEG/JPEG 2000 aware client. Besides, the JPSearch File Format library supports the extraction of attached metadata descriptions of a JPSearch File Format based file. The extracted metadata can then be further processed.
Installation Guideline	Runs as Java application or can be used as Java library in projects .Detailed information are provided below.
Interface Description	See below.
INPUT	Encoding: 1 to n XML based metadata descriptions (currently MPEG-7, Dublin Core and JPSearch Core is supported), an image file in JPEG, JPEG 2000, PGM (raw), PPM (raw) or PGX format and basic annotation information as specified in the JPSearch Part 4 standard.

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

© ISO/IEC 2012 – All rights reserved

	Decoding: An image file encoded in JPSearch File Format (JPEG or JPEG 2000)
OUTPUT	Encoding: A JPEG or JPEG 2000 encoded image file including the given XML based metadata descriptions. Decoding: All integrated metadata descriptions are extracted from the input file and stored at the local file system.
Programming Language(s)	Java 1.6 and higher
Platform(s)	Any platform with a Java installation
Dependencies	Java 1.6, JPEG/JPEG 2000 encoder/decoder
Details	See below

5.3.2 Architecture

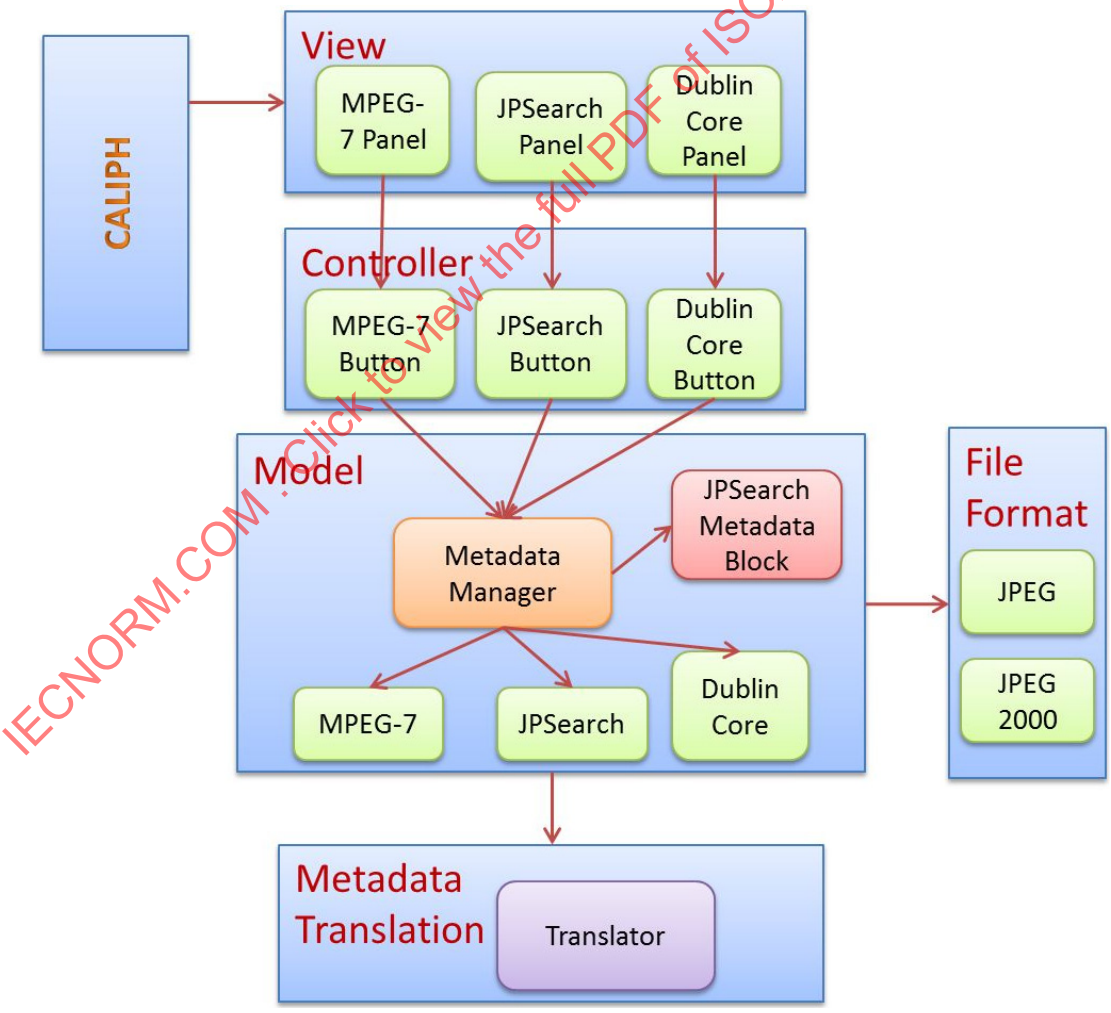

Figure 4 — Architecture of the JPSearch File Format application

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

© ISO/IEC 2012 – All rights reserved

Figure 4 presents the architecture of the JPSearch File Format framework. The implementation bases on the following external technologies:Java Swing for the graphical user interface and JDOM for manipulating XML instance documents. The framework supports two different usage modes. On the one side, it offers an API providing means for encoding and decoding image files to and from the JPSearch File Format which can be used in various projects. The API is implemented in the Model part of the architecture and in more detail by the `MetadataManager` class. This class can be used by other projects. Furthermore, for demonstration purposes a standalone version has been provided which usage is described below (see 5.3.3).

In detail, the Model component provides the implementation for the functionality of decoding, encoding, metadata assignment and translation. The main element in this component is the `MetadataManager` class. This class is implemented as singleton and ensures consistency among the instantiated metadata descriptions. The translation between metadata formats is realized by the reference software of Part 2 (see 5.1 for detailed information). The current implementation supports three metadata formats, namely *translateFromJPSearch* (to map to JPSearch Core), *translateFromMPEG7* (to map to MPEG-7) and *translateFromDC* (to map to Dublin Core). Furthermore, the `MetadataManager` supports checking the validity of the created XML instance documents (by using the *`checkValdidation()`* method). By calling the *`doImport()`* method, the `MetadataManager` allows the creation of a valid image file concerning to the JPSearch File Format specification. Besides, by using the *`doExport()`* method a JPSearch File Format file can be decoded and the contained information is extracted.

On the other side, the framework has been integrated as a use case into the well known image annotation tool Caliph (http://www.semanticmetadata.net/features/). Here, the Module component has been extended by two additional parts, namely the View and Controller component.

The View component enhances Caliph by specialized user interfaces that provide means for adding metadata annotations in the supported metadata formats, namely MPEG-7, Dublin Core and JPSearch Core. Example screenshots of the respective interfaces are provided in Figure 5 (for MPEG-7), Figure 6 (for Dublin Core) and Figure 7 (for JPSearch Core).

The Controller component delegates the individual functionality requests (save, translate, delete, etc.) from the respective view elements to the model component. Details on how to use the extended Caliph tool can be found in 5.3.4.

5.3.3 Installation / Utilization

The standalone version comes as java library which can be used via a console or instantiated in a larger Java application. The two main scenarios are encoding of metadata into images and decoding of the images in order to receive the metadata elements.

5.3.3.1 Encoding of Images

The encoding of images and its assigned metadata can be accomplished by executing the de.global.Main class. The class provides a set of parameters for configuration which are summarized in the following:

-i_encode <file name>

name of the input file + format ending. Supported formats are PGM (raw), PPM

(raw), PGX, JPEG and JPEG 2000

-o <file name>

name of the output file with ending .jp2 or jpeg

-auth <author´s name>

name of the author, "given name" and "family name" where both information is

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

© ISO/IEC 2012 – All rights reserved

required

-conf <confident measure>

only number in the range of [1..15] is allowed

-flag [yes|no]

flag read only.

-jp (optional)

JPSearch XML file. (.xml ending required)

-mpeg (optional)

MPEG-7 XML file. (.xml ending required)

-dc (optional)

Dublin Core XML file. (.xml ending required)

-trans [jp|mpeg|dc] (optional)

Metadata format from which the translation starts.

If no metadata exists, jp2 file won´t be created

A possible call might be look like as follows:

java –i_encode iris.ppm -o iris.jp2 -auth Joe Doe -conf 7 -flag no -mpeg mpeg7eval.mp7.xml -dc dublincoreTest.xml -trans jp de.portal.Main

5.3.3.2 Decoding of Images

Decoding simply is activated by using the following parameter:

-i_decode <file name>

name of the input file. it has to be a file with JP2 or JPEG format.

This results in a decoded image and the attached metadata format files of the JPSearch file. Those XML documents (currently JPSearch Core, Dublin Core and MPEG-7) are stored at the local file system. Furthermore, the console shows information regarding the annotation part of the JPSearch file format.

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

© ISO/IEC 2012 – All rights reserved

5.3.4 Demonstration

This Subclause describes a demonstration of an implementation of the JPSearch File Format which supports the storage of metadata descriptions in a JPEG 2000 coded image and the translation among metadata formats. The provided implementation provides an adaptation of the well known Caliph software for additional metadata annotation and translation between metadata formats. By this, the individual metadata descriptions in the arbitrary metadata formats can be stored as JPSearch File Format in a JPEG 2000 coded image file.

The JPSearch File Format framework comes with an extended Caliph user interface supporting in addition Dublin Core and JPSearch Core.

5.3.4.1 Visualization of MPEG-7

The original Caliph implementation already provided visual components for annotating metadata with the MPEG-7 format. However, as the tool has been extended by additional metadata formats, the MPEG- 7 view has been organized within a tabbed pane (see Figure 5). Besides, the original implementation already provided means for annotation visual effects of images (e.g. color) and semantic descriptions. Furthermore, the pane allows through the buttons Show XML and Save XML, the visualization of the XML content as well as the storage of the active metadata description into a separate file at the local file system.

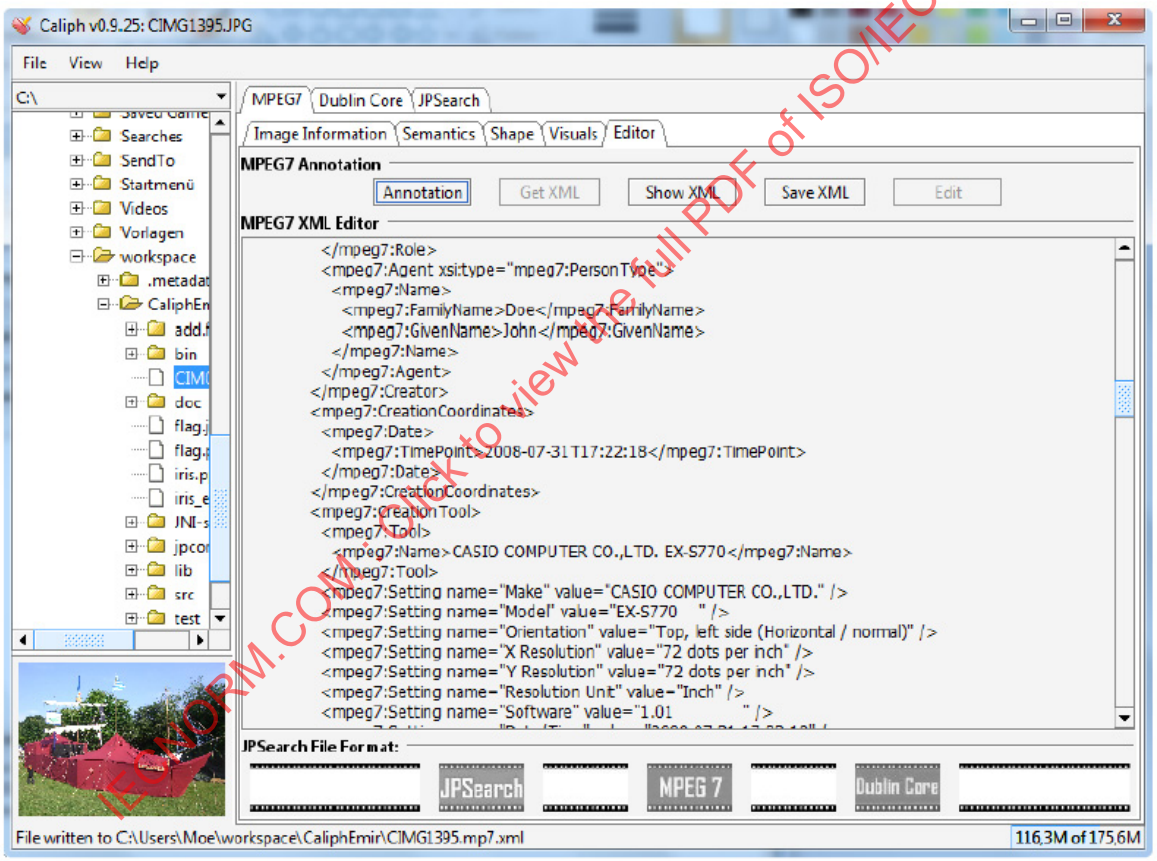

Figure 5 — Screenshot for MPEG-7 annotations in the extended Caliph application

5.3.5 Visualization of Dublin Core

Similar to the MPEG-7 visualisation, an annotation module for the Dublin Core schema has been integrated (see Figure 6). As the Dublin Core schema is focused on 15 core elements, two tabs are sufficient. Besides, the manual annotation in the provided text fields, a user is able to load preconfigured annotations by using the

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

© ISO/IEC 2012 – All rights reserved

Get XML button. Again, the Save XML functionality supports the storage of the annotation in an external XML instance document at the file system.

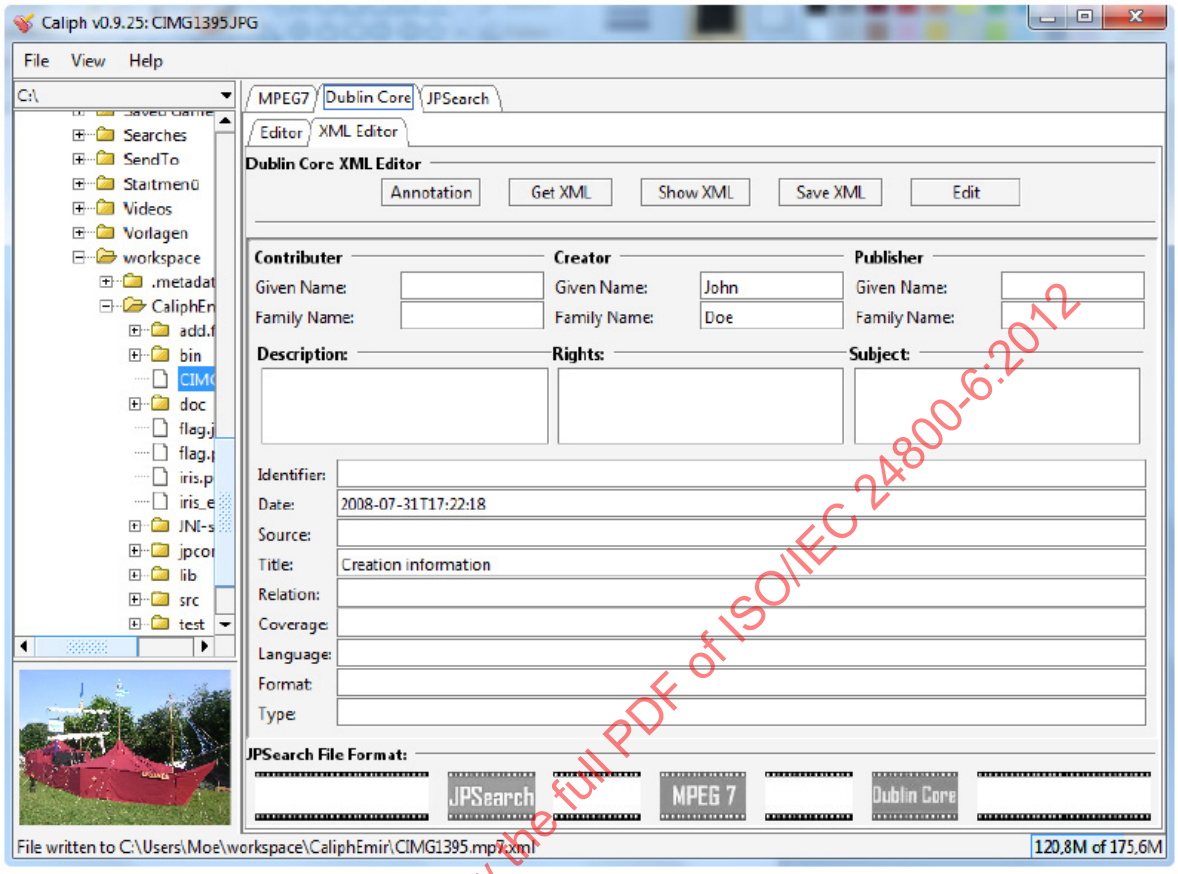

Figure 6 — Screenshot for Dublin Core annotations in the extended Caliph application

5.3.6 Visualization of JPSearch Core

Finally, the current implementation offers a user interface for annotating metadata according to the JPSearch core metadata format (see Figure 7). For supporting the full standard, 3 tabs for manually annotating the metadata are integrated. Here, again the full functionality (Get XML, Show XML, Save XML) for managing the description is available.

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

© ISO/IEC 2012 – All rights reserved

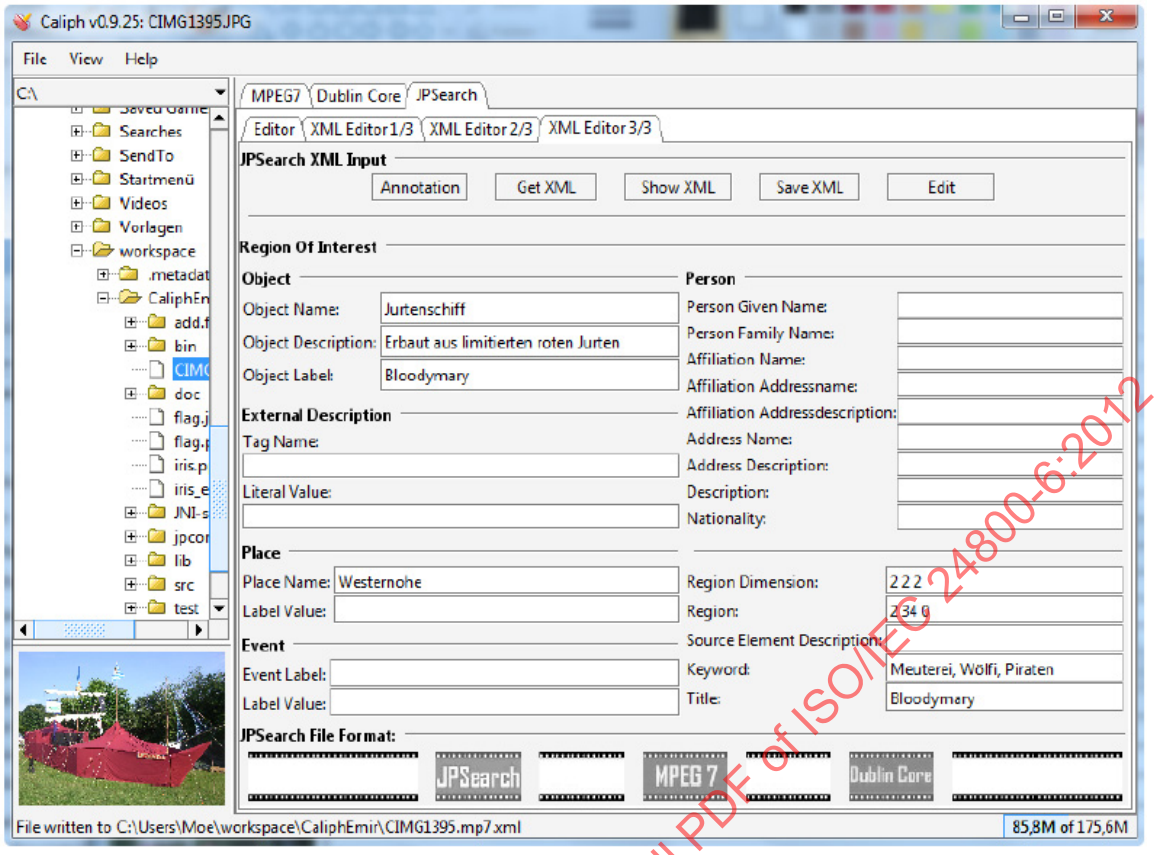

Figure 7 — Screenshot for JPSearch core annotations in the extended Caliph application

5.3.7 Translation Rules

The framework uses internally for translating metadata instances the translation rules framework introduced in 5.1. The workflow of any translation among the metadata instances is realized by an intermediate translation to JPSearch core and then the target schema is addressed (see Figure 8).

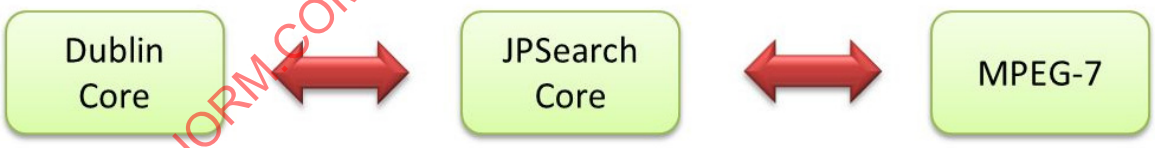

Figure 8 — Metadata translation directions

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

5.4 Repository Import/Export Module

5.4.1 Summary

Module name	Repository Import/Export Module
Functional Description	Content Repositories import/export

© ISO/IEC 2012 – All rights reserved

Installation Guideline	Copy the Java jar file to the local filesystem. Requires a previous Java 1.6 installation and the operating system execution path environment variable properly pointing to the Java application launcher (java command).
Interface Description	*java -Dlog4j.configuration=file:./WEB-INF/classes/log4j.properties -classpath ./WEB-INF/lib/mpqf-1.0.jar;./WEB-INF/lib/xmldb.jar;./WEB-INF/lib/exist.jar;./WEB-INF/lib/log4j-1.2.15.jar;./WEB-INF/lib/xmlrpc-1.2-patched.jar;./WEB-INF/lib/jaxen-1.1.1.jar;./WEB-INF/lib/commons-pool-1.4.jar;./WEB-INF/lib/antlr-2.7.6.jar;./WEB-INF/lib/xercesImpl-2.9.1.jar;./WEB-INF/lib/resolver-1.2.jar;./WEB-INF/lib/quartz-1.6.0.jar;./WEB-INF/lib/commons-logging-1.0.4.jar;./WEB-INF/lib/jta.jar;./WEB-INF/lib/commons-collections-3.1.jar;./WEB-INF/lib/stax-api-1.0.1.jar;./WEB-INF/lib/caliph-emir-cbir.jar;./WEB-INF/lib/lucene-core-2.1.0.jar;./WEB-INF/lib/lire.jar* ***org.barcelonatech.JPSearchImportTester [test_jpsearch_p5_interchangefile.xml] [repositoryname]***
INPUT	An XML file to be imported / export request
OUTPUT	A report about the import process / An exported XML file
Programming Language(s)	Java
Platform(s)	Windows, Linux and any other platform supporting Java 1.6.
Dependencies	NO
Details	Standalone tool which allows the import of a single ISO/IEC 24800-5 metadata file containing the description of multiple images.

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

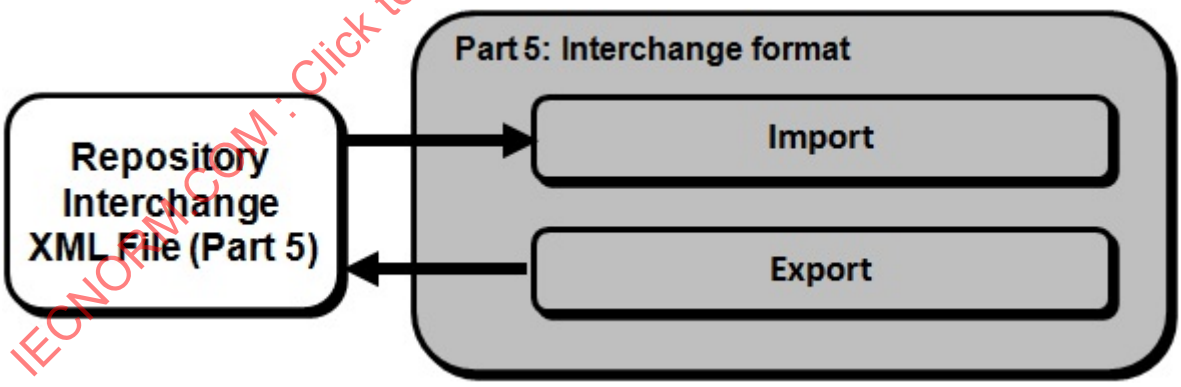

Figure 9 — Repository contents import/export module architecture overview

© ISO/IEC 2012 – All rights reserved

5.4.2 Functionality

The table below lists the features which are covered by the provided software.

Feature	Description
Import XML	JPSearch Part 5 XML Format Import.
Export XML	JPSearch Part 5 XML Format Export.

5.4.3 Command line utilization

5.4.3.1 Instruction for commands

This module provides a standalone tool which allows the import of a single ISO/IEC 24800-5 metadata file containing the description of multiple images. The same module provides functionalities for Parts 3 and 5, so the results of the import/export tests can be also queried with Part 3 tools as explained in 5.2.

The module comes as a Java jar file and relies on a Java 1.6 (or higher) installation on the target computer.

The provided software allows importing or exporting metadata against one or more image metadata DBs stored in directories (one for each DB) within the *imageDB_index* directory. Initially, there only exists one empty DB called "blank", which allows executing Part 3 queries against a non persistent metadata DB generated from an input XML file (this DB should not be used to test the import/export tools). The import tool allows creating a new DB and filling it with the contents from an input 24800-5:2011 XML file. A newly created DB can be manually removed by just deleting its directory from the imageDB_index directory. The export tool allows storing into an output 24800-5:2011 XML file the contents of a previously created DB.

IECNORM.COM : Click to view the full PDF of ISO/IEC 24800-6:2012

IMPORT TOOL

The standalone test version of the import tool can be executed by the following command:

java -Dlog4j.configuration=file:./WEB-INF/classes/log4j.properties -classpath ./WEB-INF/lib/mpqf-1.0.jar;./WEB-INF/lib/xmldb.jar;./WEB-INF/lib/exist.jar;./WEB-INF/lib/log4j-1.2.15.jar;./WEB-INF/lib/xmlrpc-1.2-patched.jar;./WEB-INF/lib/jaxen-1.1.1.jar;./WEB-INF/lib/commons-pool-1.4.jar;./WEB-INF/lib/antlr-2.7.6.jar;./WEB-INF/lib/xercesImpl-2.9.1.jar;./WEB-INF/lib/resolver-1.2.jar;./WEB-INF/lib/quartz-1.6.0.jar;./WEB-INF/lib/commons-logging-1.0.4.jar;./WEB-INF/lib/jta.jar;./WEB-INF/lib/commons-collections-3.1.jar;./WEB-INF/lib/stax-api-1.0.1.jar;./WEB-INF/lib/caliph-emir-cbir.jar;./WEB-INF/lib/lucene-core-2.1.0.jar;./WEB-INF/lib/lire.jar ***org.barcelonatech. JPSearchImportTester [test_jpsearch_p5_interchangefile.xml] [repositoryname]***

A .bat/.sh script which instantiates this command with two parameters is provided within the ***test/jpsearch_tests*** directory:

metadataImport.bat ***[test_jpsearch_p5_interchangefile.xml] [repositoryname]***

And the version without parameters:

test_metadataImport_ex1.bat

EXPORT TOOL

The standalone test version of the export tool can be executed by the following command:

java -Dlog4j.configuration=file:./WEB-INF/classes/log4j.properties -classpath ./WEB-INF/lib/mpqf-1.0.jar;./WEB-INF/lib/xmldb.jar;./WEB-INF/lib/exist.jar;./WEB-INF/lib/log4j-1.2.15.jar;./WEB-INF/lib/xmlrpc-1.2-patched.jar;./WEB-INF/lib/jaxen-1.1.1.jar;./WEB-INF/lib/commons-pool-1.4.jar;./WEB-INF/lib/antlr-

© ISO/IEC 2012 – All rights reserved