JHOVE2: Next-Generation Architecture for Format-Aware Characterization
XML Module
Version 0.4
Issued 2010-02-21
Status Draft

1 Introduction

JHOVE2 is a framework and application for next-generation format-aware characterization of digital objects. The function of JHOVE2 is encapsulated in a series of modules that can be configured for use within the framework’s plug-in architecture. The XML module provides characterization services for the Extensible Markup Language family of format.

---

**Important information for users of the JHOVE2 XML module**

The authoritative specification for XML [[REC-xml-20081126] is unambiguous.

Validation of XML instances by this module is comprehensive.

**NOTE** A format specification is considered unambiguous if there is broad community consensus regarding the intention of all normative requirements of the format’s authoritative specification; otherwise it is considered ambiguous, and areas of potential ambiguity will be documented below.

Module validation is considered comprehensive if all normative requirements defined by that specification are validated by the module; otherwise it is considered selective, and non-validated features will be documented below.

---

2 Identification

<table>
<thead>
<tr>
<th>Primary format or format family</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Canonical format name:</td>
<td>XML</td>
</tr>
<tr>
<td>Alias format name(s):</td>
<td>Extensible Markup Language (XML) 1.0</td>
</tr>
<tr>
<td>Canonical format identifier:</td>
<td>JHOVE2 info:jhove2/format/xml</td>
</tr>
<tr>
<td>Alias format identifier(s):</td>
<td>PRONOM PUID: fmt/101</td>
</tr>
<tr>
<td>MIME</td>
<td>application/xml</td>
</tr>
<tr>
<td></td>
<td>application/xml-external-parsed-entity</td>
</tr>
<tr>
<td></td>
<td>application/xml-dtd</td>
</tr>
<tr>
<td></td>
<td>text/xml (deprecated)</td>
</tr>
<tr>
<td></td>
<td>text/xml-external-parsed-entity (deprecated)</td>
</tr>
<tr>
<td></td>
<td>[application</td>
</tr>
<tr>
<td>RFC</td>
<td>RFC 3023</td>
</tr>
<tr>
<td></td>
<td>draft-murata-kohn-lilley-xml-04</td>
</tr>
<tr>
<td>UTI</td>
<td>public.xml</td>
</tr>
<tr>
<td></td>
<td>(Apple Uniform Type Identifier)</td>
</tr>
</tbody>
</table>
### JHOVE2 XML module

<table>
<thead>
<tr>
<th>JHOVE2 module name:</th>
<th>XmlModule</th>
</tr>
</thead>
<tbody>
<tr>
<td>JHOVE2 module identifier:</td>
<td>JHOVE2 info:jhove2/reportable/org/jhove2/module/format/XmlModule</td>
</tr>
</tbody>
</table>
| JHOVE2 module class | org.jhove2.module.format.xml.XmlModule.java  
org.jhove2.module.format.xml.XmlModule.class |
| JHOVE2 module jar |  |

### 3 References

For the purposes of the JHOVE2 XML module the authoritative format specifications are:

- **[REC-xml-20081126]** *Extensible Markup Language (XML) 1.0 (Fifth Edition)*  
  W3C Recommendation 26 November 2008  
  [http://www.w3.org/TR/2008/REC-xml-20081126](http://www.w3.org/TR/2008/REC-xml-20081126)

- **[REC-xml11-20060816]** *Extensible Markup Language (XML) 1.1 (Second Edition)*  
  W3C Recommendation 16 August 2006, edited in place 29 September 2006  
  [http://www.w3.org/TR/2006/REC-xml11-20060816](http://www.w3.org/TR/2006/REC-xml11-20060816)

The following XML specifications are considered informative:

  [http://www.w3.org/XML/](http://www.w3.org/XML/)

  [http://www.w3.org/XML/Core/#Publications](http://www.w3.org/XML/Core/#Publications)

- **[REC-xml-names-20091208]** *Namespaces in XML 1.0 (Third Edition)* 8 December 2009  

- **[REC-xml-names11-20060816]** *Namespaces in XML 1.1 (Second Edition)* 16 August 2006  
  [http://www.w3.org/TR/2006/REC-xml-names11-20060816](http://www.w3.org/TR/2006/REC-xml-names11-20060816)

- **[XML Conformance]** *Extensible Markup Language (XML) Conformance Test Suites*  
  [http://www.w3.org/XML/Test](http://www.w3.org/XML/Test)

- **[XML Schema]** *XML Schema*  
Other Useful References

Tim Bray. The Annotated XML Specification. 1998
http://www.xml.com/axml/testaxml.htm

http://www.snee.com/bob/xmlann


Kenneth B. Sall. XML Family of Specifications: A Practical Guide. 2002
http://www.wdvl.com/Authoring/Languages/XML/XMLFamily
http://kensall.com/big-picture/bigpix22.html

Uche Ogbuji. A survey of XML standards

W3Schools Online Web Tutorials
http://www.w3schools.com/default.asp

http://www.datypic.com/books/defxmlschema/

Links regarding the MIME types for XML

MIME type "text/xml" is now generally considered to be depreciated. MIME type "application/xml" is preferred so that the character set encoding will be read from the xml header. See:
http://www.w3.org/2006/02/son-of-3023/draft-murata-kohn-lilley-xml-04.html
http://www.ietf.org/rfc/rfc3023.txt
http://www.iana.org/assignments/media-types/

Discussions as to why application/xml is favored over text/xml
http://www.imc.org/ietf-xml-mime/mail-archive/msg00978.html
http://xml.coverpages.org/xmlMediaMIME.html.
http://www.xml.com/cs/user/view/cs_msg/2124

Links regarding Uniform Type Identifier

4 Terminology and conventions

The terminology used to describe XML documents is defined in the body of the [XML] specification.

3rd party glossaries
http://www.developer.com/xml/article.php/2106671
http://www.studio6.ca/xml_glossary.html

5 Validity

XML validity will be reported in terms of the two status values defined by [XML]:

- A well-formed XML instance satisfies the syntax rules provided in the specification
- A valid XML instance has an associated document type declaration (DTD) or Schema and complies with the structural and content constraints expressed in that declaration.

References:
http://www.w3schools.com/Xml/xml_dtd.asp
http://www.xml.com/pub/a/98/10/guide0.html?page=4

6 Format profiles

JHOVE2 treats the XML format as a family having only one profile. Character set encoding and XML version number will be treated as reportable properties of the format, instead of defining a separate profile.

"XML 1.1 updates XML so that it no longer depends on the specific Unicode version... It also adds checking of normalization, and follows the Unicode line ending rules more closely."

"Whereas XML 1.0 provided a rigid definition of names, wherein everything that was not permitted was forbidden, XML 1.1 names are designed so that everything that is not forbidden (for a specific reason) is permitted. Since Unicode will continue to grow past version 4.0, further changes to XML can be avoided by allowing almost any character, including those not yet assigned, in names."

References regarding XML 1.0 versus XML 1.1
http://www.digitalpreservation.gov/formats/fdd/fdd000264.shtml
7 Reportable properties

The list below does not exhaustively traverse the hierarchy of all possible properties which may be reported by the XML Module. In most cases only the top-level reportable properties annotated within a given class is reported.

<table>
<thead>
<tr>
<th>Property</th>
<th>Identifier</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XmlModule</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>SaxParser</td>
<td>SaxParser</td>
<td>XML Parser name, features, properties</td>
</tr>
<tr>
<td>Identifier</td>
<td><a href="http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/SaxParser">http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/SaxParser</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>XmlDeclaration</td>
<td>XmlDeclaration</td>
<td>XML Declaration data</td>
</tr>
<tr>
<td>Identifier</td>
<td><a href="http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/XmlDeclaration">http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/XmlDeclaration</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>RootElement</td>
<td>RootElement</td>
<td>The document's root element</td>
</tr>
<tr>
<td>Identifier</td>
<td><a href="http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/RootElement">http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/RootElement</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>DTDs</td>
<td>List&lt;DTD&gt;</td>
<td>List of Document Scope Definitions (DTDs)</td>
</tr>
<tr>
<td>Identifier</td>
<td><a href="http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/DTDs">http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/DTDs</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>NamespaceInformation</td>
<td>NamespaceInformation</td>
<td>XML Namespace Information</td>
</tr>
<tr>
<td>Identifier</td>
<td><a href="http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/NamespaceInformation">http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/NamespaceInformation</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Entities</td>
<td>List&lt;Entity&gt;</td>
<td>List of Entity Declarations</td>
</tr>
<tr>
<td>Identifier</td>
<td><a href="http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/Entities">http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/Entities</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>EntityReferences</td>
<td>ArrayList&lt;EntityReferences$EntityReference&gt;</td>
<td>List of Entity References</td>
</tr>
<tr>
<td>Identifier</td>
<td><a href="http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/EntityReferences">http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/EntityReferences</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Notations</td>
<td>List&lt;Notation&gt;</td>
<td>List of Notations found in the XML document</td>
</tr>
<tr>
<td>Identifier</td>
<td><a href="http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/Notations">http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/Notations</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>NumericCharacterReferences</td>
<td>ArrayList&lt;NumericCharacterReferences$NumericCharacterReference&gt;</td>
<td>List of Numeric Character References</td>
</tr>
<tr>
<td>Identifier</td>
<td><a href="http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/NumericCharacterReferences">http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/NumericCharacterReferences</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>ProcessingInstructions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identifier</td>
<td><a href="http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/ProcessingInstructions">http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/ProcessingInstructions</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>List&lt;ProcessingInstruction&gt;</td>
<td>List of Processing Instructions</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Property**
- **Identifier**: http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/CommentInformation
- **Type**: CommentInformation
- **Description**: List of Comments

**Property**
- **Identifier**: http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/ValidationResults
- **Type**: ValidationResults
- **Description**: Warning or error messages generated during XML Validation

**Property**
- **Identifier**: http://jhove2.org/terms/property/org/jhove2/module/format/xml/XmlModule/isWellFormed
- **Type**: boolean
- **Description**: XML well-formed status

**Message**
- **Identifier**: http://jhove2.org/terms/message/org/jhove2/module/format/xml/XmlModule/FailFast
- **Type**: Message
- **Description**: Fail fast message.

**Message**
- **Identifier**: http://jhove2.org/terms/message/org/jhove2/module/format/xml/XmlModule/SaxParserMessages
- **Type**: List<Message>
- **Description**: SAX Parser Messages.

**Validator**
- **Property**
  - **Identifier**: http://jhove2.org/terms/property/org/jhove2/module/format/Validator/isValid
  - **Type**: Validator$Validity
  - **Description**: Validation status.

**Property**
- **Identifier**: http://jhove2.org/terms/property/org/jhove2/module/format/Validator/Coverage
- **Type**: Validator$Coverage
- **Description**: Format module validation coverage.

**BaseFormatModule**
- **Message**
  - **Identifier**: http://jhove2.org/terms/message/org/jhove2/module/format/BaseFormatModule/ModuleNotFoundMessage
  - **Type**: Message
  - **Description**: Format Module Not Found Error Message

**Message**
- **Identifier**: http://jhove2.org/terms/message/org/jhove2/module/format/BaseFormatModule/ModuleNotFormatModuleMessage
- **Type**: Message
- **Description**: Module returned is not Format Module Error Message

**FormatModule**
- **Property**
  - **Identifier**: http://jhove2.org/terms/property/org/jhove2/module/format/FormatModule/Format
  - **Type**: Format
  - **Description**: Format module format.
<table>
<thead>
<tr>
<th>Property</th>
<th>Identifier</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReleaseDate</td>
<td><a href="http://jhove2.org/terms/property/org/jhove2/module/Module/ReleaseDate">http://jhove2.org/terms/property/org/jhove2/module/Module/ReleaseDate</a></td>
<td>String</td>
<td>Module release date.</td>
</tr>
<tr>
<td>Scope</td>
<td><a href="http://jhove2.org/terms/property/org/jhove2/module/Module/Scope">http://jhove2.org/terms/property/org/jhove2/module/Module/Scope</a></td>
<td>Module$Scope</td>
<td>Module Type generic or specific (to a source unit.)</td>
</tr>
<tr>
<td>Note</td>
<td><a href="http://jhove2.org/terms/property/org/jhove2/module/Module/Note">http://jhove2.org/terms/property/org/jhove2/module/Module/Note</a></td>
<td>String</td>
<td>Module informative note.</td>
</tr>
<tr>
<td>TimerInfo</td>
<td><a href="http://jhove2.org/terms/property/org/jhove2/module/Module/TimerInfo">http://jhove2.org/terms/property/org/jhove2/module/Module/TimerInfo</a></td>
<td>TimerInfo</td>
<td>Timer info for this module.</td>
</tr>
</tbody>
</table>

**SaxParser**

<table>
<thead>
<tr>
<th>Property</th>
<th>Identifier</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parser</td>
<td><a href="http://jhove2.org/terms/property/org/jhove2/module/format/xml/SaxParser/Parser">http://jhove2.org/terms/property/org/jhove2/module/format/xml/SaxParser/Parser</a></td>
<td>String</td>
<td>Java class used to parse the XML.</td>
</tr>
<tr>
<td>SaxFeatures</td>
<td><a href="http://jhove2.org/terms/property/org/jhove2/module/format/xml/SaxParser/SaxFeatures">http://jhove2.org/terms/property/org/jhove2/module/format/xml/SaxParser/SaxFeatures</a></td>
<td>List&lt;String&gt;</td>
<td>SAX Parser Feature Settings</td>
</tr>
</tbody>
</table>
8 Configuration

As stated in the JHOVE2 Architectural Overview:

JHOVE2 uses the Spring Java enterprise platform (www.springsource.org) to manage module instantiation. JHOVE2 is implemented in terms of the Spring philosophy of dependency injection: all relationships between JHOVE2 components are defined in external configuration files, not the Java source code, so they can be modified easily by a JHOVE2 user at the time of installation or invocation.

Spring Config File
The Spring configuration file for the XmlModule and the XML Format is config/spring/module/format/xml/jhove2-xml-config.xml

The Spring bean with id="XmlModule" is used to create instances of the module.

This bean among other things, instantiates the values of 3 properties that can be customized to affect the parsing behavior of this module.

The property with name="saxParser" is used to specify a reference to another Spring bean in the same config file, which is used to initialize the SAX2 Parser used by the XML Module. See below.

The property with name="ncrParser" is used to set a true/false flag that specifies whether the XML module should perform a second parsing operation (in addition to the standard SAX2 parse), that extracts and reports the numeric character references that may be contained in the XML document being characterized.

The property with name="collectCommentText" is used to set a true/false flag that specifies whether (possibly voluminous) comment text should be saved by the SAX2 parser for later reporting.

SAX Parser Specification
The "SaxParser" bean in the Spring config file is used to explicitly specify the parser to be used and the customized behaviors that the parser should follow.

The property named "parser" specified the classname of the SAX2 XMLReader implementation to be used for parsing XML instances. As delivered, the value of this property is explicitly set to: org.apache.xerces.parsers.SAXParser

which specifies use of the Xerces2 Java Parser. (http://xerces.apache.org/xerces2-j), whose library files are packaged as part of the JHOVE2 installation.

If this property value is omitted, then the Parser class name will be determined using the logic of
XMLReader org.xml.sax.helpers.XMLReaderFactory.createXMLReader()
as documented at
http://www.saxproject.org/apidoc/org/xml/sax/helpers/XMLReaderFactory.html
(In this scenario, the value of the environmental variable org.xml.sax.driver is used if it has been set, or
the value can be read from the META-INF/services/org.xml.sax.driver file of a jar file in the
classpath.)

Note that the XmlModule code requires a parser that implements the SAX2 core and Extensions 1.1 (such
as the enhanced DeclHandler, LexicalHandler, and Locator2 interfaces). If a less capable parser is used,
the module will either fail or only be able to report a restricted set of characterization information.

SAX Feature Settings

The SaxParser Spring bean also allows the JHOVE2 user to customize many behaviors of the SAX2
parser. The standard set of features is documented at the SAX Project web site:
http://www.saxproject.org/get-set.html
http://www.saxproject.org/apidoc/org/xml/sax/package-summary.html#package_description

Additional Xerces2 SAX optional features are documented at
http://xerces.apache.org/xerces2-j/features.html

Note that a given parser might not support all the possible feature options.
In the Spring config file for the module as delivered, the following features are set:

   Validation should occur when the XML instance is parsed.
   http://xml.org/sax/features/validation = true

   Supress reporting of absolutized System IDs in entity declarations (relative to base URI)
   http://xml.org/sax/features/resolve-dtd-uris = false

   Validate the document only if a grammar is specified
   http://apache.org/xml/features/validation/dynamic = true

   Turn on XML Schema validation
   http://apache.org/xml/features/validation/schema = true

   Enable full schema grammar constraint checking
   http://apache.org/xml/features/validation/schema-full-checking = false

   Use only the first schema location hint encountered by the processor to locate the
   components for a given target namespace
   http://apache.org/xml/features/honour-all-schemaLocations = false

XML Catalog

If the "useXmlCatalog" property of the SaxParser is set to true, then the XML Catalog files listed
within the "xmlCatalogList" property will be used for external entity resolution. This allows the
JHOVE2 user to alter the normal default lookup behavior when trying to resolve System
Identifiers, Public Identifiers, or Schema Locations.
As currently implemented, the names of the catalog files must be specified as absolute paths. In future we will allow use of directory paths that are relative to the config folder.

9 Implementation Notes

Validating Parsers

The XmlModule has been implemented using the SAX2 API with SAX Extensions 1.1. SAX provides a mature parser API, with a smaller memory requirement than DOM2. The StAX2 API was considered, but not chosen as its enhancements were not needed for JHOVE2).

The remainder of this section more fully documents the various parser choices that were researched and provides links to references that document those APIs.

The software component used to read a XML instance while testing conformance to a DTD or Schema is known as a validating parser. The abstraction layer provided by Java 6 is known as JAXP (the Java API for XML Processing). JAXP provides the interfaces and abstract classes that 3rd party products build upon to implement the actual parsing functionality. JAXP 1.4 specifies the following parser alternatives:

- **DOM2** (Document Object Model) uses a tree-oriented mechanism that reads the entire XML document into memory. The abstract class `javax.xml.parsers.DocumentBuilder` defines the API that a DOM2 parser should implement. The abstract class `javax.xml.parsers.DocumentBuilderFactory` defines the `newDocumentBuilder` method, which can be used to create a parser instance. A commonly used implementation of `DocumentBuilderFactory` is `org.apache.xerces.jaxp.DocumentBuilderFactoryImpl` from the Apache Xerces project, which creates an instance of `org.apache.xerces.jaxp.DocumentBuilderImpl`.
  
  DOM2 provides random access to any portion of the document once it has been parsed. This is useful for searching the tree using XPath queries, for XSLT transforms, and for in-place edits of the document. But this flexibility comes at the expense of a large memory footprint.

- **SAX2** (Simple API for XML) uses a streaming push mechanism. The class `org.xml.sax.XMLReader` declares the interface that a SAX2 parser must implement. The class `org.xml.sax.helpers.XMLReaderFactory` provides the `createXMLReader` method, which method should be supplied the name of a parser that implements the interface, such as `org.apache.xerces.parsers.SAXParser` from the Apache Xerces-project.

  SAX2 has a much smaller memory requirement and typically executes faster. XML data is "pushed" or streamed into the application, which processes it in a forward-only manner. The application interface is event-driven requiring a content handler to handle each of the different events of interest (e.g. start of document, start of element, end of element). Writing SAX-based parser code, however, is more difficult than writing DOM code. The current state (such as location in the document tree) must be maintained by the code that you write.

- **StAX2** (Streaming API for XML) uses a streaming pull mechanism that has higher performance than DOM, and is easier to use than SAX. StAX offers both a cursor and an iterator API. The
class javax.xml.stream.XMLStreamReader specifies the cursor interface. Class javax.xml.stream.XMLEventReader specifies the iterator interface. The Apache Geronimo team chose the Woodstox implementation of these interfaces for performance reasons.

StAX is similar to SAX in that it does not build an in-memory document tree, and reads the document in a forward-only manner. Unlike SAX, StAX implements a "pull" mechanism whereby the application asks the parser engine for more data, instead of having to operate in a reactive mode. Thus a StAX parser is generally easier to work with than SAX, without sacrificing efficiency. The Woodstox XML processor appears to be the best choice of StAX implementations.

JAXP:
https://jaxp.dev.java.net/
http://www.xml.com/pub/a/2005/07/06/jaxp.html
http://codeidol.com/java/java-xml/
http://www.edankert.com/validate.html

Xerces2:
http://xerces.apache.org/xerces2-j/

DOM2:
http://www.w3.org/TR/DOM-Level-2-Core/
http://java.sun.com/j2ee/1.4/docs/tutorial/doc/JAXPDOM8.html

SAX2:
http://www.saxproject.org/
http://www.cafeconleche.org/books/xmljava/chapters/ch06.html
http://java.sun.com/javase/6/docs/api/org/xml/sax/XMLReader.html
http://java.sun.com/javase/6/docs/api/org/xml/sax/helpers/XMLReaderFactory.html

StAX2:
http://www.vsj.co.uk/articles/display.asp?id=643
http://java.sun.com/webservices/docs/1.6/tutorial/doc/SJSXP.html
http://www.xmlpull.org/
http://en.wikipedia.org/wiki/StAX
http://woodstox.codehaus.org/

XML Catalog
References to external entities such as DTDs or Schemas can be overridden by the use of an XML Catalog which provides a mapping between those references and locally-cached equivalents. A parser that supports the XML Catalog mechanism can use this mapping when validating an XML instance.

The JHOVE2 XML Module utilizes the Xerces2 XMLCatalogResolver

http://xerces.apache.org/xerces2-j/faq-xcatalogs.html
which has a dependency on the Apache XML Commons resolver
http://xml.apache.org/commons/components/resolver/

References defining the OASIS XML Catalog Standard:

http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=entity
OASIS Entity Resolution TC
XML Catalogs OASIS Standard V1.1, 7 October 2005
http://www.oasis-open.org/committees/entity/faq.php

References that are informative with regard to XML Catalog

Writings by Norman Walsh

http://java.sun.com/webservices/docs/1.6/jaxb/catalog.html
XML Entity and URI Resolvers Version 1.1 by Norman Walsh 2001
http://norman.walsh.name/2007/02/06/xmlresolver
Building a better resolver Volume 10, Issue 8; 06 Feb 2007
http://norman.walsh.name/2007/02/14/resolvers
All your resolvers are belong to us Volume 10, Issue 12; 14 Feb 2007

Writings by Elliotte Rusty Harold

http://www.cafeconleche.org/books/effectivexml/chapters/47.html
Effective XML 2003
Managing XML data: XML catalogs 2005

The Java API reference

http://xerces.apache.org/xerces2-j/faq-xcatalogs.html
Using XML Catalogs
http://www.xml.com/pub/a/2004/03/03/catalogs.html
Using XML Catalogs with JAXP by Tom White 2004

Wikipedia