



THE
le@rning
FEDERATION

schools online curriculum content initiative

LEARNING OBJECT REPOSITORY ACCESS AND EXCHANGE Web Service Specification

(LORAX4)

VERSION: 4.01

DATE: 1 JUNE 2009

DOCUMENT: SOAP_SPECIFICATION_V4.01.DOC

<http://www.thelearningfederation.edu.au/standards>

Amendment History

Date	Issue	Comment	Person
15/07/2002	0.1	Document Created	Nigel Ward Ari Stathopolous Dytech Solutions
25/07/2002	0.2	Aligned search and result terms with IMS learning object metadata. Removed Query Thesaurus parameter from QueryLO. Simplified permissible search parameters. Removed QueryThesaurus method from WSDL Clarified search ranking and case sensitivity	Nigel Ward Ari Stathopolous Dytech Solutions Simon Sinnott Nicola Pitkanen
31/07/2002	0.3	Fine-tuned WSDL. Named protocol LORAX. Added Changes under Consideration section.	Nigel Ward Dytech Solutions Released to The Learning Federation ECC for comment.
24/01/2003	0.4	Updated Monitoring and Usage section with possible future changes to the specification	Nigel Ward
20/02/2007	4.0	Added LORAX 4 changes Significant rewrite and addition to document.	Paul Fell
1/06/2009	4.1	Illustrated configuration variables for retrieving content in ANZ-LOM metadata format.	Paul Fell Brett Munro Steve Sunter

Approvals

Name/Organisation/Position	Signature	Date
Nick Weidemann Chief Technology Officer Curriculum Corporation		
Preety Agarwal Senior Manager The Le@rning Federation		

Table of Contents

1	INTRODUCTION.....	1
	1.1 Purpose.....	1
	1.2 Obligations of the Le@rning Federation	1
	1.3 Monitoring and usage.....	1
	1.4 Conformance.....	2
2	PRINCIPLES	2
3	OVERVIEW OF SOAP OPERATIONS	2
	3.1 Query Content (QueryContent).....	3
	3.2 Retrieve Content (RetrieveContent).....	3
	3.3 Retrieve Content Metadata (RetrieveContentMetadata)	4
	3.4 Retrieve Content Rights (RetrieveContentRights)	5
	3.5 Test Connection (TestConnection).....	5
4	SEARCH PARAMETERS	5
	4.1 Criteria/ Query Structure	5
	4.2 Search Properties	6
	4.3 Examples	7
	REFERENCES	9

1 Introduction

Schools access The Le@rning Federation's online educational content within a framework of distributed access to education system Learning Object gateways. The Le@rning Federation provides access to online educational content via a repository called 'the Exchange'. Education systems can retrieve online educational content from the Exchange and provide distribution through their online systems. The education systems also provide manipulation tools and e-learning environments required by schools.

1.1 Purpose

The Le@rning Federation's Learning Object Repository Access and eXchange (LORAX) Specification defines a web service for interacting with the Exchange repository of learning objects. The web service provides a simple programmatic interface allowing client systems to discover and download metadata, intellectual property rights and learning objects from the Exchange.

1.2 Obligations of the Le@rning Federation

Curriculum Corporation and the shareholders will fulfil the following obligations in undertaking and delivering The Le@rning Federation outcomes.

- to consult with relevant organisations and user groups in establishing, implementing and reviewing this Exchange web service specification.
- to proactively review and, as appropriate, evolve the Exchange web service specification to reflect knowledge and practice derived from implementation of the Exchange and interfacing the Exchange with external systems.
- to contribute to national and international standards development in the area of Digital Repository Interfaces through the evolving body of knowledge and practice generated under the auspices of the project.

1.3 Monitoring and usage

The LORAX Specification defines the web service interface to the Exchange for discovering and downloading learning objects and metadata at the time of publication.

The specification will be updated and enhanced during development and deployment of the Exchange. Updated specifications and related guidelines will be documented and published on the TLF website <http://www.thelearningfederation.edu.au>.

Previous LORAX specifications

This is the fourth release specification. LORAX 4 is backwards compliant with both LORAX 3 and LORAX 2 specifications. That is, any system utilising LORAX 2 or 3 will be able to use all implemented methods within LORAX 4.

Note: LORAX 4 is not backwards compliant with LORAX 1 – that protocol is no longer supported by TLF.

LORAX history

LORAX 1 – The first LORAX specification.

LORAX 2 – Addition of support for content of types "Resources"

- "QueryLO" method became "QueryContent",
- "DownloadLO" method became "RetrieveContent",

- “RetrieveLOMetadata” became “RetrieveContentMetadata”.

LORAX 3 – Addition of “RetrieveContentRights” method.

LORAX 4 – Addition of ANZ-LOM and SCORM content support.

Potential future modifications

This is a fourth release specification. It is anticipated that at times, the specification may change after consultation with The Learning Federation stakeholders.

Identifying versions of the web service

The version of the LORAX web service being used is indicated via year and month components within the targetNamespace URL in the WSDL definition of the web service. For example, the targetNamespace used within this specification is

<http://lex.thelearningfederation.edu.au/webservices/2006/12/LORAX.asmx>. This indicates that the web service is defined using the WSDL published in December 2006.

New versions of the WSDL will be given new targetNamespaces that include year and month components.

1.4 Conformance

The Exchange has implemented all the web service operations described in this specification.

Client systems conforming to this Specification must implement the QueryContent and RetrieveContent operations described in this specification. Client system implementation of the RetrieveContentMetadata and TestConnection and RetrieveContentRights operations is optional.

2 Principles

Development of the Exchange SOAP Interface Specification is underpinned by the following principles and goals:

- Simple discovery of learning objects, including
 - Ability to discover new learning objects
- Use of standard metadata and learning object packaging technologies.

3 Overview of SOAP Operations

The SOAP interface consists of five operations:

- Query Content (QueryContent)
- Retrieve Content (RetrieveContent)
- Retrieve Content Metadata (RetrieveContentMetadata)
- Retrieve Content Rights (RetrieveContentRights)
- Test Connection (TestConnection)

For a detailed definition of these operations and the complete web service refer to the Web Services Description Language (WSDL) definition in Appendix C.

3.1 Query Content (QueryContent)

The Query Content (QueryContent) operation takes a search request and returns short descriptions of published learning objects that match the search. The search request is a structured XML query described in Section 4.

The operation expects the following parameters:

Parameter	Description
Criteria	Query string criteria formatted as described in Section 4, essentially an XML document containing the search parameters
Username	A valid Exchange username
Password	A valid Exchange password

The operation returns a sequence of short descriptions of matching published content contained in a <SearchResult> XML structure:

Element	Description
<SearchResult>	Short description of a matching content. Contains <Title>, <Location>, <VersionDate>, <Catalog> and <Identifier>, elements.
<Title>	Title of the content item.
<Location>	URL for the content item.
<VersionDate>	The published date of content being returned.
<Catalog>	Catalog of "TLF-LearningObject" or "TLF-Resource" allowing a search to be limited over a subset of content. E.g. an "L" or "R" prefix to an Identifier.
<Identifier>	Identifier for the content item. Content is identified using positive integers.

The SearchResult structure is detailed in Appendix A. The Criteria parameter must consist of an XML document containing the search parameters as defined in Appendix B. This is discussed in the following section.

3.2 Retrieve Content (RetrieveContent)

The Retrieve Content (RetrieveContent) operation retrieves the Learning Object or Resource specified by the content identifier, allowing the packaging and rights formats to be specified. The content is returned as zipped IMS Package Interchange Files (IMS Content Packaging Specification, 2001). The rightsDigitalFormat parameter should be passed as an empty string ("") to specify the default encoding.

The operation expects the following parameters:

Parameter	Description
Identifier	An Integer corresponding to content identifier.. A Learning Object or Resource identifier ('Lxxx' or 'Rxxx')
packageFormat	A packaging format identifier. The packageFormat parameter should be passed "IMS CP 1.1.4 + ANZ-LOMv1.0", "SCORM 1.3", or "" (to specify the default packaging). The default packaging format is the superseded TLF MAP custom metadata format "IMS 1.1.3 + Metadata

	1.2.4". Example syntax: <code>download.setPackageFormat("IMS CP 1.1.4 + ANZ-LOMv1.0");</code>
rightsDigitalFormat	A rights format identifier. The rightsDigitalFormat parameter should be passed as an empty string ("") to specify the default encoding.
Username	A valid Exchange username
Password	A valid Exchange password

The operation returns a piece of content encoded as zipped binary data according to the IMS Content Packaging Specification.

Only published content can be downloaded from the Exchange. If the download request identifies unpublished or non-existing learning objects, an error condition is returned.

3.3 Retrieve Content Metadata (RetrieveContentMetadata)

The Retrieve Content Metadata (RetrieveContentMetadata) operation takes a sequence of learning object identifiers and returns complete metadata descriptions for those pieces of content. The metadata is returned as XML formatted according to the IMS metadata XML binding (IMS Learning Resource Meta-Data XML Binding, 2001).

The operation expects the following parameters:

Parameter	Description
Identifier	A Learning Object or Resource identifier ('Lxxx' or 'Rxxx'). Each <Identifier> element contains a positive integer.
metadataFormat	A metadata format identifier.
Username	A valid Exchange username
Password	A valid Exchange password

The return value is an XML document containing the Learning Object or Resource metadata. These XML documents conform to The Exchange metadata XML Schema and the corresponding Metadata Application Profile.

The metadataFormat parameter should be passed "IMS 1.2.4", or "" (to specify the default encoding). The default metadata format is "IMS 1.2.4". The operation returns a sequence of metadata descriptions for the content. The result of the operation is a sequence of <lom> XML elements, formatted according to the IMS metadata XML binding.

Element	Description
<lom>	Metadata for the content, encoded according to the IMS Metadata XML binding.

Only metadata for published content can be downloaded from the Exchange. If the download request identifies unpublished or non-existing content, an error condition is returned.

3.4 Retrieve Content Rights (RetrieveContentRights)

The Retrieve Content Rights (RetrieveContentRights) operation retrieves the user's rights over the Learning Object or Resource specified by the Content ID.

The operation expects the following parameters:

Parameter	Description
Identifier	A Learning Object or Resource identifier ('Lxxx' or 'Rxxx').
rightsDigitalFormat	A rights format identifier.
Username	A valid Exchange username
Password	A valid Exchange password

The return value is an XML document containing user's rights over the Learning Object or Resource. These XML documents conform to an XML Schema that corresponds to the specified digital rights format.

The rightsDigitalFormat parameter should be passed an empty string ("") to specify the default encoding.

3.5 Test Connection (TestConnection)

The Test Connection (TestConnection) operation can be used to test the web service connection without affecting any data.

The operation expects the following parameters:

Parameter	Description
Username	A valid Exchange username
Password	A valid Exchange password

If the web service connection is accessible, the result of the operation is a SOAP response with an empty body.

4 Search Parameters

The *Criteria* parameter of the QueryContent routine is used to pass all search information to the web service in a single XML document. The XML Schema in *Appendix B: XML Schema* for QueryString specifies the structure of this parameter but in addition there are semantic rules that must be understood to use the search correctly.

4.1 Criteria/ Query Structure

The *Criteria* parameter of the QueryContent routine is used to pass all search information to the web service in a single XML document. The XML Schema in *Appendix B: XML Schema* for QueryString specifies the structure of this parameter but in addition there are semantic rules that must be understood to use the search correctly.

- Child nodes of an AND node are logically conjoined (ANDed together.)
- Sibling AND nodes are logically conjoined (ANDed together.)
- Child nodes of an OR node are logically disjointed (ORed together.)

For example the pseudo query:

```

<QUERY>
  <AND>
    <OR>
      <A/>
      <B/>
    </OR>
  <C/>
</AND>
<AND>
  <D/>
</AND>
</QUERY>

```

is equivalent to the Boolean statement ((A OR B) AND C) AND D.

4.2 Search Properties

Within the <AND> and <OR> structure the remaining elements (<Title>, <Description>, etc.) specify properties of the content that must be matched by the search. The permissible content for each of these elements varies, as does the way in which they are interpreted by the search engine. For instance the <Identifier> element is used to specify the integer object identifier of the file or content within the Exchange, whereas the <Title> element can contain a Boolean string similar to the advanced search expressions of popular web search engines. The following table lists the meaning and applicable format for each element.

XML Element	Definition	Permissible Values
Title, Description	The content title and description.	A text search expression, containing literal words, literal phrases (in quotes) and Boolean operators AND and OR. This syntax is discussed below. Note: Either node can be used to find matches in both content titles and descriptions. The two nodes remain to maintain compatibility with earlier versions of LORAX.
VersionDate	The (UTC) date upon which the current published version was published.	A date in the form YYYY-MM-DD as specified in the Schema. The node must also contain an attribute "op" with a value "<", ">", "=", "<=" or ">=" also as specified in the Schema. This operator specifies the relationship between the document date and the specified date. For instance the node <pre><VersionDate op="&gt;">2004-01-01</VersionDate></pre> returns all documents with a version date after January 1 st 2004.
Identifier	The Exchange's unique identifier for content.	A valid Exchange learning object or resource handle. Handles are positive integers as specified in the Schema. <pre><Identifier>270</Identifier></pre> returns the learning object L270. Note: This node has little practical value, but remains to maintain compatibility with earlier versions of LORAX. As such, the identifier must be specified as "xxxx", not as "Lxxxx" or "Rxxxx".

XML Element	Definition	Permissible Values
Catalog, Catalogue	The Exchange's learning objects are catalogued separately to resources. This node can be used to specify a catalogue.	One of the following two options: <code><Catalog>TLF-LearningObject</Catalog></code> Returns learning objects only. <code><Catalog>TLF-Resource</Catalog></code> Returns resources only.

The <Title> and <Description> elements can contain Boolean strings similar to the syntax used by popular web search engines. This is interpreted as a free-text search on the corresponding field and may contain the keywords AND and OR as well as parentheses to build a complex Boolean condition. The search criteria in these elements have the following features:

- The keywords AND and OR may be used to construct Boolean search conditions.
- Parentheses may be used to group conditions.
- The search string is case insensitive and in particular the keywords are case insensitive.
- To distinguish the actual words *and* and *or* from the Boolean keywords they must be enclosed in double quotes (usually within a phrase). For instance:
`<Title>"Blue and Green"</Title>`
 will match the literal string *Blue and Green*.
- Single quotes are treated as normal characters and will be matched literally **unless** the first character and the last character in the string are both single quotes in which case the entire string will be treated as a phrase.
- Any sequence of words not containing the Boolean keywords AND or OR is automatically treated as a phrase. Therefore it is never necessary to use double quotes unless you wish to explicitly use the words *and* or *or* in a phrase.

4.3 Examples

The following XML fragments demonstrate valid Criteria parameters.

Example 1.

All published content within The Exchange.

```
<QUERY>
  <OR>
    <Catalog>TLF-LearningObject</Catalog>
    <Catalog>TLF-Resource</Catalog>
  </OR>
</QUERY>
```

Example 2.

All published learning objects within The Exchange.

```
<QUERY>
  <Catalog>TLF-LearningObject</Catalog>
</QUERY>
```

Example 3.

All published resources within The Exchange

```
<QUERY>
  <Catalog>TLF-Resource</Catalog>
</QUERY>
```

Example 4.

All content where the latest published version was created on or after the 12th of January 2006.

```
<QUERY>
  <VersionDate op="&gt;=">2006-01-12</VersionDate>
</QUERY>
```

Example 5.

All content with title or description containing “smoke” where the latest published version was created between the 10th of January 2003 and the 20th of July 2003.

```
<QUERY>
  <AND>
    <OR>
      <Title>Smoke</Title>
    </OR>
    <VersionDate op="&gt;=">2003-01-10</VersionDate>
    <VersionDate op="&lt;=">2003-07-20</VersionDate>
  </AND>
</QUERY>
```

Example 6.

All learning objects with title containing “Metal” or “Chrome” where the latest published version was created on or before December 31st 2002.

```
<QUERY>
  <AND>
    <OR>
      <Title>Smoke</Title>
      <Title>Chrome</Title>
    </OR>
    <VersionDate op="&lt;=">2002-12-31</VersionDate>
  </AND>
</QUERY>
```

References

IMS Content Packaging Specification, Version 1.1.4, 2004

<http://www.imsproject.org/content/packaging/>

IMS Content Packaging Specification, Version 1.1.3, 2003

<http://www.imsproject.org/content/packaging/>

IMS Learning Resource Meta-Data XML Binding, Version 1.1.2, 2001

<http://www.imsproject.org/content/packaging/>

Appendix A: Structures returned by the web service

```
struct Content
{
    string Identifier;           // ID of the Content as "Lxxxx" or "Rxxxx"
    byte[] Data;               // The Package containing the Learning Object or Resource
}
```

The client application that receives these Content structures is expected to save the Content.Data byte array to a binary disk file that has ".ZIP" file name extension.

```
struct SearchResult
{
    string Identifier;           // ID of the Content as "Lxxxx" or "Rxxxx"
    int VersionID;             // ID of the Version (to distinguish the Content's
    // version.)
    DateTime VersionDateTime; // UTC date/time the version was published in
    // W3C standard format
    string Title;              // Title of the Learning Object/Resource
    string Location;           // URL of the Learning Object/Resource
}
```

Appendix B: XML Schema for QueryString

An electronic copy of this document can be found at the following address:

<http://lex.thelearningfederation.edu.au/WebServices/2006/12/Definitions/QuerySyntax>

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace =
"http://lex.thelearningfederation.edu.au/webservices/2006/12/Definitions/QuerySyntax"
  xmlns:qs =
"http://lex.thelearningfederation.edu.au/webservices/2006/12/Definitions/QuerySyntax"
  xmlns:xs = "http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <xs:element name="QUERY" type="qs:QUERYType"/>
  <xs:element name="AND" type="qs:ANDType"/>
  <xs:element name="Identifier" type="xs:ID"/>
  <xs:element name="OR" type="qs:ORType"/>
  <xs:element name="VersionDate" type="qs:VersionDateType"/>
  <xs:element name="Title" type="xs:string"/>
  <xs:element name="Description" type="xs:string"/>
  <xs:element name="Catalog" type="xs:string"/>
  <xs:element name="Catalogue" type="xs:string"/>

  <xs:group name="SearchItemGroup">
    <xs:choice>
      <xs:element ref="qs:Identifier" maxOccurs="unbounded"/>
      <xs:element ref="qs:VersionDate" maxOccurs="unbounded"/>
      <xs:element ref="qs:Title" maxOccurs="unbounded"/>
      <xs:element ref="qs:Description" maxOccurs="unbounded"/>
      <xs:element ref="qs:Catalog" maxOccurs="unbounded"/>
      <xs:element ref="qs:Catalogue" maxOccurs="unbounded"/>
    </xs:choice>
  </xs:group>

  <xs:complexType name="QUERYType">
    <xs:sequence>
      <xs:element ref="qs:AND" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="ANDType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:group ref="qs:SearchItemGroup"/>
      <xs:element ref="qs:OR"/>
    </xs:choice>
  </xs:complexType>

```

```
</xs:complexType>

<xs:complexType name="ORType">
  <xs:group ref="qs:SearchItemGroup" maxOccurs="unbounded" />
</xs:complexType>

<xs:simpleType name="VersionDateRegex">
  <xs:restriction base="xs:string">
    <xs:pattern value="[0-9]{4}-[0-9]{2}-[0-9]{2}" />
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="VersionDateType">
  <xs:simpleContent>
    <xs:extension base="qs:VersionDateRegex">
      <xs:attribute name="op">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="&lt;" />
            <xs:enumeration value="&gt;" />
            <xs:enumeration value="=" />
            <xs:enumeration value="&lt;=" />
            <xs:enumeration value="&gt;=" />
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

</xs:schema>
```

Appendix C: WSDL for the web service

An electronic copy of this document can be found at the following address:

<http://lex.thelearningfederation.edu.au/webservices/2006/12/LORAX/WSDL>

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions
  xmlns:s1="http://lex.thelearningfederation.edu.au/webservices/2006/12/Definitions/QuerySyntax"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tns="http://lex.thelearningfederation.edu.au/webservices/2006/12/LORAX"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  targetNamespace="http://lex.thelearningfederation.edu.au/webservices/2006/12/LORAX"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
      targetNamespace="http://lex.thelearningfederation.edu.au/webservices/2006/12/LORAX">
      <s:import
        namespace="http://lex.thelearningfederation.edu.au/webservices/2006/12/Definitions/QuerySyntax" />
      <s:element name="TestConnection">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1"
              name="username" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1"
              name="password" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="TestConnectionResponse">
        <s:complexType />
      </s:element>
      <s:element name="RetrieveContent">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1"
              name="contentIdentifier" type="s:ID" />
            <s:element minOccurs="0" maxOccurs="1"
              name="packageFormat" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1"
              name="rightsDigitalFormat" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1"
              name="username" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1"
              name="password" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="RetrieveContentResponse">

```

```

        <s:complexType>
            <s:sequence>
                <s:element minOccurs="1" maxOccurs="1"
name="RetrieveContentResult" type="tns:Content" />
            </s:sequence>
        </s:complexType>
    </s:element>
    <s:complexType name="Content">
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1"
name="Identifier" type="s:ID" />
            <s:element minOccurs="0" maxOccurs="1"
name="Data" type="s:base64Binary" />
        </s:sequence>
    </s:complexType>
    <s:element name="RetrieveContentMetadata">
        <s:complexType>
            <s:sequence>
                <s:element minOccurs="0" maxOccurs="1"
name="contentIdentifier" type="s:ID" />
                <s:element minOccurs="0" maxOccurs="1"
name="metadataFormat" type="s:string" />
                <s:element minOccurs="0" maxOccurs="1"
name="username" type="s:string" />
                <s:element minOccurs="0" maxOccurs="1"
name="password" type="s:string" />
            </s:sequence>
        </s:complexType>
    </s:element>
    <s:element name="RetrieveContentMetadataResponse">
        <s:complexType>
            <s:sequence>
                <s:element minOccurs="0" maxOccurs="1"
name="lom">
                    <s:complexType mixed="true">
                        <s:sequence>
                            <s:any />
                        </s:sequence>
                    </s:complexType>
                </s:element>
            </s:sequence>
        </s:complexType>
    </s:element>
    <s:element name="RetrieveContentRights">
        <s:complexType>
            <s:sequence>
                <s:element minOccurs="0" maxOccurs="1"
name="contentIdentifier" type="s:ID" />
                <s:element minOccurs="0" maxOccurs="1"
name="rightsDigitalFormat" type="s:string" />
                <s:element minOccurs="0" maxOccurs="1"
name="username" type="s:string" />

```

```

        <s:element minOccurs="0" maxOccurs="1"
name="password" type="s:string" />
    </s:sequence>
</s:complexType>
</s:element>
<s:element name="RetrieveContentRightsResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1"
name="RetrieveContentRightsResult">
                <s:complexType mixed="true">
                    <s:sequence>
                        <s:any />
                    </s:sequence>
                </s:complexType>
            </s:element>
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="QueryContent">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1"
ref="s1:QUERY" />
            <s:element minOccurs="0" maxOccurs="1"
name="username" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1"
name="password" type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="QueryContentResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0"
maxOccurs="unbounded" name="QueryContentResult" type="tns:SearchResult" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:complexType name="SearchResult">
    <s:sequence>
        <s:element minOccurs="0" maxOccurs="1"
name="Identifier" type="s:ID" />
        <s:element minOccurs="1" maxOccurs="1"
name="VersionID" type="s:int" />
        <s:element minOccurs="1" maxOccurs="1"
name="VersionDateTime" type="s:dateTime" />
        <s:element minOccurs="0" maxOccurs="1"
name="Title" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1"
name="Location" type="s:string" />
    </s:sequence>

```

```

        </s:complexType>
    </s:schema>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://lex.thelearningfederation.edu.au/webservices/2006/12/Definitions/QuerySyntax">
        <s:element name="QUERY">
            <s:complexType mixed="true">
                <s:sequence>
                    <s:any />
                </s:sequence>
            </s:complexType>
        </s:element>
    </s:schema>
</wsdl:types>
<wsdl:message name="TestConnectionSoapIn">
    <wsdl:part name="parameters" element="tns:TestConnection" />
</wsdl:message>
<wsdl:message name="TestConnectionSoapOut">
    <wsdl:part name="parameters" element="tns:TestConnectionResponse" />
</wsdl:message>
<wsdl:message name="RetrieveContentSoapIn">
    <wsdl:part name="parameters" element="tns:RetrieveContent" />
</wsdl:message>
<wsdl:message name="RetrieveContentSoapOut">
    <wsdl:part name="parameters" element="tns:RetrieveContentResponse" />
</wsdl:message>
<wsdl:message name="RetrieveContentMetadataSoapIn">
    <wsdl:part name="parameters" element="tns:RetrieveContentMetadata" />
</wsdl:message>
<wsdl:message name="RetrieveContentMetadataSoapOut">
    <wsdl:part name="parameters"
element="tns:RetrieveContentMetadataResponse" />
</wsdl:message>
<wsdl:message name="RetrieveContentRightsSoapIn">
    <wsdl:part name="parameters" element="tns:RetrieveContentRights" />
</wsdl:message>
<wsdl:message name="RetrieveContentRightsSoapOut">
    <wsdl:part name="parameters"
element="tns:RetrieveContentRightsResponse" />
</wsdl:message>
<wsdl:message name="QueryContentSoapIn">
    <wsdl:part name="parameters" element="tns:QueryContent" />
</wsdl:message>
<wsdl:message name="QueryContentSoapOut">
    <wsdl:part name="parameters" element="tns:QueryContentResponse" />
</wsdl:message>
<wsdl:portType name="LORAXSoap">
    <wsdl:operation name="TestConnection">

```

```

        <documentation xmlns="http://schemas.xmlsoap.org/wsdl/">This can
be used to test the web service connection without affecting any data.</documentation>
        <wsdl:input message="tns:TestConnectionSoapIn" />
        <wsdl:output message="tns:TestConnectionSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="RetrieveContent">
        <documentation
xmlns="http://schemas.xmlsoap.org/wsdl/">Retrieves the Learning Object or Resource
specified by the 'identifier', allowing the format of the digital rights to be
specified.</documentation>
        <wsdl:input message="tns:RetrieveContentSoapIn" />
        <wsdl:output message="tns:RetrieveContentSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="RetrieveContentMetadata">
        <documentation
xmlns="http://schemas.xmlsoap.org/wsdl/">Retrieves the metadata for the content
specified by 'identifier'.</documentation>
        <wsdl:input message="tns:RetrieveContentMetadataSoapIn" />
        <wsdl:output message="tns:RetrieveContentMetadataSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="RetrieveContentRights">
        <documentation
xmlns="http://schemas.xmlsoap.org/wsdl/">Retrieves the rights for the content
specified by 'identifier'.</documentation>
        <wsdl:input message="tns:RetrieveContentRightsSoapIn" />
        <wsdl:output message="tns:RetrieveContentRightsSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="QueryContent">
        <documentation
xmlns="http://schemas.xmlsoap.org/wsdl/">Retrieves an array of SearchResult's given an
(XML) Criteria search term.</documentation>
        <wsdl:input message="tns:QueryContentSoapIn" />
        <wsdl:output message="tns:QueryContentSoapOut" />
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="LORAXSoap" type="tns:LORAXSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
style="document" />
    <wsdl:operation name="TestConnection">
        <soap:operation
soapAction="http://lex.thelearningfederation.edu.au/webservices/2006/12/LORAX/TestConn
ection" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RetrieveContent">
        <soap:operation
soapAction="http://lex.thelearningfederation.edu.au/webservices/2006/12/LORAX/Retrieve
Content" style="document" />

```

```

        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RetrieveContentMetadata">
        <soap:operation
            soapAction="http://lex.thelearningfederation.edu.au/webservices/2006/12/LORAX/Retrieve
            ContentMetadata" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RetrieveContentRights">
        <soap:operation
            soapAction="http://lex.thelearningfederation.edu.au/webservices/2006/12/LORAX/Retrieve
            ContentRights" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="QueryContent">
        <soap:operation
            soapAction="http://lex.thelearningfederation.edu.au/webservices/2006/12/LORAX/QueryCon
            tent" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="LORAX">
    <documentation xmlns="http://schemas.xmlsoap.org/wsdl/">The LORAX
    (Learning Object Repository Access and eXchange) soap interface. (September 2006
    Edition)</documentation>
    <wsdl:port name="LORAXSoap" binding="tns:LORAXSoap">
        <soap:address
            location="http://lex.thelearningfederation.edu.au/WebServices/2006/12/LORAX.asmx" />
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```