SRU Server

We are developing an SRU server for Lucene, based on the OCLC SRU implementation. When used with the Lucene indices created by Fedora gSearch or our own fedora indexing service, this server provides two advantages over the VTLS Fedora SRU server: it is more configurable, and it can interact directly with Lucene (rather than Fedora's default search index).

We can test the current functionality of the server with the SRU Server Tester.

Eventually, the server will support CQL level 2.

OSU server

Peter Krenesky at Oregon State's Open Source Lab has developed a similar server called SRWLucene. We want to coordinate with this development as much as possible.

SRU server base URLs

Base URL	Purpose
http://fedora.dlib.indiana.edu:8080/SRW/search/FedoraAdmin?	PURL resolution
http://fedora.dlib.indiana.edu:8080/SRW/search/GSearch?	Fedora objects
http://fedora.dlib.indiana.edu:8080/SRW/search/INHarm?	IN Harmony objects
http://brie.dlib.indiana.edu:8080/SRW/search/DSpace?	DSpace objects
http://brie.dlib.indiana.edu:8080/SRW/search/DSpaceDark?	Restricted DSpace objects
MANY more...	Collection-specific

Sample SRU queries

- Simple test to make sure the server is running:
  http://fedora.dlib.indiana.edu:8080/SRW/search/test?query=dog
- Title contains 'road' (any collection)
  http://fedora.dlib.indiana.edu:8080/SRW/search/GSearch?query=dc.title%3Droad
- Title contains 'road' and 'cabin' (any collection)
  http://fedora.dlib.indiana.edu:8080/SRW/search/GSearch?query=dc.title%3D%22road%20and%20cabin%22
- All Fedora objects that belong to the ihs/sheetmusic collection
  http://fedora.dlib.indiana.edu:8080/SRW/search/GSearch?query=iudl.collection%3D%22ihs%2Fsheetmusic%22
- Title contains 'woman' and collection is Hoagy
  http://fedora.dlib.indiana.edu:8080/SRW/search/GSearch?query=dc.title%3Dwoman%20and%20iudl.collection%3D%22hoagy%22
- Identifier is /hoagy/ATM-MC2-3-2-1 and we want iudlAdmin results
  http://fedora.dlib.indiana.edu:8080/SRW/search/GSearch?query=dc.identifier+%3D+%22%2Fhoagy%2FATM-MC2-3-2-1%22&recordSchema=iudlAdmin
- Title=road and subject="Log cabins"
- Date is between 1946 and 1948
  date within "1946 1948"
- Last 5 changes to the production fedora repository http://fedora.dlib.indiana.edu:8080/SRW/search/FedoraAdmin?query=cql.allRecords+%3D+%221%22&version=1.1&operation=searchRetrieve&recordSchema=info%3Ahttp%3A%2F%2Fdlib.indiana.edu%2Fxml%2FiudlAdmin%2Fversion1.0%2F&maximumRecords=5&startRecord=1&resultSetTTL=300&recordPacking=xml&recordXPath=&sortKeys=foxml.lastModifiedDate,,0,,lowValue

Implementation details

CQL queries are parsed, then converted into Lucene QueryParser syntax, which is then forwarded to Lucene. When Lucene returns Hits, they are converted into a simple XML fragment that is placed into the SRU Response object.

Basic SRU features

Explain: Fully supported

Search/Retrieve:

Param	Req?	Status
version	Y	always returns 1.1
query	Y	works
startRecord	N	works
maximumRecords	N	works
recordPacking	N	should work
recordSchema	N	works
recordXPath	N	not supported
resultSetTTL	N	works
sortKeys	Y	Some support: index and ascending/descending, and missingValue, lowValue, highValue are supported but all else is ignored.
stylesheet	N	not supported
extraRequestData	Y	facet information
operation	Y	works

Scan:

Scan is currently disabled, because it was having problems, and we don't need it yet.

Search/Retrieve Response

The format of the response objects is dictated by a configuration file. Any number of recordSchemas can be created, using any combination of fields from the Lucene index.

Our default response format will be MODS.

Do we want to start including the recordIdentifier in our result objects?

CQL features

Relations

Relation	Supported?	Notes
=	Y	
<	N	
>	N	
<=	N	
>=	N	
<>	N	
scr	Y	same as =
exact	Y	Added for IN Harmony and only currently works on that database. Will eventually work on all.
all	Y	
any	Y	
within	Y	Works, but has unexpected behavior if performed on fields with multiple values, or that stores data in unusual ways (TOKENIZED). Support will be added to make this work **better** and to make it work as expected for dates.
encloses	N	should be implicit for our date searches

Booleans

Boolean	Supported?	Notes
and	Y	
or	Y	
not	Y	
prox	N	will need for text collections, must define supported modifiers

Modifiers

We don't have an immediate need for modifiers, but we will eventually want to support much of the bibliographic profile.

Wildcards, etc.

We need to support wildcards. Anchors can be ommitted for now.

CQL context set

field	support	details
resultSetId	yes	Support for re-fetching a resultset is optimized, and support for combining a resultset with others or with other search clauses is supported.
serverChoice index	supported	
allRecords	supported	
anyIndexes	supported	

Relation support is noted above (in general support).

None of the modifiers/qualifiers are supported.

DC context set

All of the DC context set is supported.

Bib context set

None of the bib context set is supported, but we will gradually start using this as needed.

Sorting

Do we want to support the new sortby feature? (This is more practical than the old XPath entry in the regular SRU parameters). What types of sort do we need?

Installing the server

Always delete the old copy from Tomcat, compile with "ant deploy" and restart Tomcat. **Don't try to deploy from Eclipse. It won't work!** After the WAR Tomcat expands the WAR file, you will need to edit these config files:

- SRWServer.props
    - include the correct Tomcat directory in SRW.Home and index.html
    - place the directory of the Lucene index in db.GSearch.home
- GSearch.SRWDatabase.props
    - edit the mapping from the CQL index set to your Lucene fields

    - edit the supported result schemas and the mapping from Lucene fields to these schemas

Upgrading the server

Simply copy the SRW.jar, SRWLucene.jar and any modified configuration files to the server while it's been stopped and then start it.

Changes to OCLC code

- added ORG.oclc.os.SRW.Lucene.LuceneDatabase
- added ORG.oclc.os.SRW.Lucene.LuceneSearchResult
- edited build.xml
- edited SRWServer.props
- added GSearch.SRWDatabase.props
- **need to check on this!** I had changed SRWDatabaseImpl and QueryResult to pass schemaName, but I think they are now back to the original state.

Java flow-of-control

SRWServlet

- doGet
    - SRWServletInfo reads the config files and sets up properities for all "known" databases.
    - passes off to processMethodRequest
        - SRWServletInfo.setSRWStuff parses the DB out of the request string,
            - SRWDatabase initializes a database object
                - Instantiates the proper class (as listed in SRWServer.props)
                - Reads in properties from the DB's config file
        - looks up some DB details, and sets them as properties of msgContext
    - Builds an SRW SOAP query out of the URL query and invokes it with AxisEngine.invoke()
    - when it returns, strips the SOAP stuff out
- doPost
    - very similar to doGet, resulting in an AxisEngine.invoke()

srw_bindings.SRWSoapBindingImpl

- searchRetrieveOperation is invoked by Axis
- eventually calls SRWDatabase.doRequest() on the proper database class

Databases derive from SRWDatabaseImpl. This class implements the doRequest() method. It is best not to override this method, as it takes care of caching result sets and other useful administrative stuff. It is best to only override the abstract methods.

Open questions

1. Lucene has some internal support for dates. Can we get gSearch to recognize this, and store the values appropriately? How to handle "truncated" dates? How important is this?