Tasks

type	key	summary	assignee	reporter	priority	status	resolution	created	updated	due

⚠ Can't show details. Ask your admin to add this Jira URL to the allowlist.

View these issues in Jira

Task Name	Time Est. hours	% Done	Assignee	Link to section
Setup Development Environment	8	0		1
Get URIs for Institution	32	0		2
Get data for an individual URI	32	0		3
Mockup of Search UI	24	0		4
Create Solr Doc from data for URI	40	0		5
Working search UI prototype	40	0		6
baisc multi-node Hadoop cluster on IaaS	40	0		7
automated and scripted cluster on IaaS	40	0		8
Data validation code for Institution's data	80	0		9
Update system	40	0		#10

1 Setup development environment

Git repository. Just copy the useful parts over from the DuraSpaceMultiSiteSearch branch of https://github.com/vivo-project/Linked-Data-Indexer .

Document single node Hadoop setup.

Development Solr service setup. Must use Sorl 4.x or greater (4.2 is the current release of Solr as of 2013-03). There have been huge improvements in Solr/Lucene going from 3.x to 4.x. I've encountered systems where setting up solr can be a bit of a chore because the instructions don't make it clear what version of solr to use and what additional libraries to add. I suggest one ofthe following 1) making the instructions very clear about which version of solr to use OR 2) automating the build by downloading a URL, and copying files to the correct location for the solr home directory.

Ant/Ivy build script. (DONE in DuraSpaceMultiSiteSearch)

Wiki/git README documentation.

2 Develop code to build list of URIs to index for Institution from standard 1.5.1 VIVO instance

There is code to parse Catalyst pages to URIs (CatalystPageToURIs.java) and to parse the JSON from VIVO (ParseDataSErviceJson.java). There is code to do the discovery of URIs for Catalyst and VIVO in LinkedDataIndexer/src/main/scal/edu/cornell/indexbuildere/discovery in VivoUriDiscoveryWorker.scala and CatalystDiscveryWorker.scala. These files could be used as examples but they depend heavily on the akka framework which we'd like to move away from.

3 Develop code to gather data required for an individual URI

See UrisForDataExpansion.java for an example of how this was done in the prototype.

4 Mockup of search UI

Base the UI for now on the current UI at vivosearch.org. Issues that will require consideration:

- whether the home-grown implementation of the responsive design (adjusting the UI in stages as screen size decreases from a full-size monitor down to low-res monitor or projector, tablet, or smartphone screen size
- how to accommodate larger numbers of institutions when a single expanded list is too long
- whether to implement additional facets beyond the current 2 (institution and type)

5 Develop code to build and index Solr document from data for URI

This depends on Mockup of the search UI in order to develop the schema for the Solr index.

SolrDocWorker.scala uses the DocumentModifier from the Vitro code to generate a Solr document from a model for a URI. We may want to reuse this approach. Much of this code is found in LinkedDataIndexer/src/main/java/edu/cornell/mannlib/vitro/webapp/search/solr. There can be found a new translate that works well without the webapp context at MultiSiteIndexToDoc.java and new DocumentModifiers that are needed for multi site indexing.

6 Working Prototype of Search UI

Make tech decisions about serving search UI and about how the UI client will communicate with the Solr service.

7 Explore multi-node Hadoop cluster deployed to IaaS

8 Scripted deploy of multi-node hadoop cluster on IaaS

9 Data Validation code for institution's data

10 Update system

Develop a system to allow updates. This is likely to involve some additional services as part of the VIVO webapp. The Mulit-site search index builder will need to query the VIVO webapp for a list of URIs that have been updated for a given time frame.