Platform X Search

# Introduction

DLAP provides a SOLR based search command that allows for searching of content in the DLAP database. Search returns a list of hit documents in relevance-ranked order. DLAP uses the SOLR search engine to create searchable indexes of item and question data, and the Search command queries against those indexes. To learn more, please see <http://gls.agilix.com/Docs/Concept/DLAPSearch>.

# Search Schema

All courses in the Platform X system have an xml schema that defines the search filter categories supported.

<bfw\_search>  
<searchcategory searchable="true" title="Ebook: Chapters, Headings, Topics" metadata="ebook" />  
<searchcategory searchable="true" title="Miscellaneous Content" metadata="other" />  
<searchcategory searchable="false" title="Exclude" metadata="exclude" />  
</bfw\_search>

Each item in the course has a meta-property in its item data section that defines the search filter to apply. All properties that are prefixed with "*meta-*" are then automatically indexed and searchable via the Agilix DLAP search command.

<meta-bfw\_searchcategory>ebook</meta-bfw\_searchcategory>

# Agilix Solr Indexing Tasks

DLAP generates some data asynchronously using background scheduled tasks. We call 2 of those tasks to kick off the Solr indexing of PX data.

**SolrIndexDomain** - Indexes all searchable content in the specified domain and optionally recurses through all descendent domains. \*This task is not scheduled and is called manually to pull in legacy data and run a full index process. It will take hours, currently 5-6, to execute.

**SolrIndexBatch** - Listens to content changed signals to index any searchable content since the last SolrIndexBatch run. This is a scheduled task currently set to run every 5 minutes.

# Importing Comments

Comments are imported into SOLR via the Data Import Request Handler.   
<http://wiki.apache.org/solr/DataImportHandler>

As defined by the apache documentation. These are the basic steps to use the handler.

* Define a data-config.xml and specify the location this file in solrconfig.xml under DataImportHandler section
* Give connection information (if you choose to put the datasource information in solrconfig)
* Open the DataImportHandler page to verify if everything is in order http://localhost:xxxx/solr/dataimport
* Use full-import command to do a full import from the database and add to Solr index
* Use delta-import command to do a delta import (get new inserts/updates) and add to Solr index

The data-config.xml is set up to point to the comments database.

<dataConfig>  
<dataSource type="JdbcDataSource" name="ds1" driver="com.microsoft.sqlserver.jdbc.SQLServerDriver"  
url="jdbc:sqlserver://VSPXDBDEV01\SQL2008R2;databaseName=PXData;responseBuffering=adaptive;"  
user="pxuser"  
password="Password1"  
readOnly="true" />

<document name="highlights">

<entity   
name="highlight" class="highlight" dataSource="ds1" pk="id"  
query = "select 'highlight|' + CAST( highlightID as varchar(50) ) as dlap\_id, courseId as dlap\_entityid,  
highlightText as dlap\_text,  
highlightText as dlap\_title,  
'Comment' as dlap\_class,  
'101' as dlap\_itemtype  
from Highlight"

transformer="DateFormatTransformer">

</entity>  
</document>  
</dataConfig>

# Executing SOLR Import for Comments

The comments import is kicked off through 3 basic http requests.

1. Remove all existing comments from index

2. Commit the comment removal

3. Execute full –import command, querying comment database and appending data to existing solr index. \*Clean=false prevents Solr from clearing out the existing index data (course items and content).

**Example requests in the DEV environment:**

http://dev.dlap.bfwpub.com:8983/solr/update?stream.body=%3Cdelete%3E%3Cquery%3Edlap\_class:Test\_Comment%3C/query%3E%3C/delete%3E

http://dev.dlap.bfwpub.com:8983/solr/update?stream.body=%3Ccommit/%3E

http://dev.dlap.bfwpub.com:8983/solr/dataimport?command=full-import&clean=false