Indexing drupal contents using Solr

|  |  |  |  |
| --- | --- | --- | --- |
| Document Type | Setup and Usage Information | | |
| **Organization** | Information Technology Systems Center | | |
| **Project** | Science on Drupal | | |
| **Originator** | ITSC | | |
| **Request #** |  | **Request Type:** | Implementation |
| **Date** | 04/16/2014 | **Client:** | ITSC Developers |
| **Prepared by** | Prabhash Jha | **Document ID:** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** | **Reviewer** |
| 04/16/2014 | 0.1 | First Draft | Prabhash Jha |  |
|  |  |  |  |  |
|  |  |  |  |  |

Table of Contents

[1. Install Solr 4](#_Toc386192838)

[a. Download latest version of solr 4](#_Toc386192839)

[b. List of steps to install solr 4](#_Toc386192840)

[2. Configure Solr instance for Drupal 4](#_Toc386192841)

[3. Drupal solr configurations 6](#_Toc386192842)

[4. Indexing Field Collections 8](#_Toc386192843)

[Original Defective Code Snapshot : 9](#_Toc386192844)

[Corrected Snapshot 10](#_Toc386192845)

[5. Implementing Spatial Filters 11](#_Toc386192846)

[a. Storing the spatial fields 11](#_Toc386192847)

[b. Enable facets 12](#_Toc386192848)

[c. Create Drupal block 12](#_Toc386192849)

[d. Input Parameters 12](#_Toc386192850)

[6. Implementing Temporal Filters: 13](#_Toc386192851)

[a. Storing Temporal Fields 13](#_Toc386192852)

[b. Enable facets 15](#_Toc386192853)

[c. Create Drupal block 15](#_Toc386192854)

[d. Input Parameters 15](#_Toc386192855)

[7. Creating Solr View 16](#_Toc386192856)

[8. Sorting solr view and modifying solr schema 17](#_Toc386192857)

[9. Creating Facets 19](#_Toc386192858)

[Facet Configuration 20](#_Toc386192859)

[Date time facets 20](#_Toc386192860)

[10. Useful Drush Commands for solr indexing 21](#_Toc386192861)

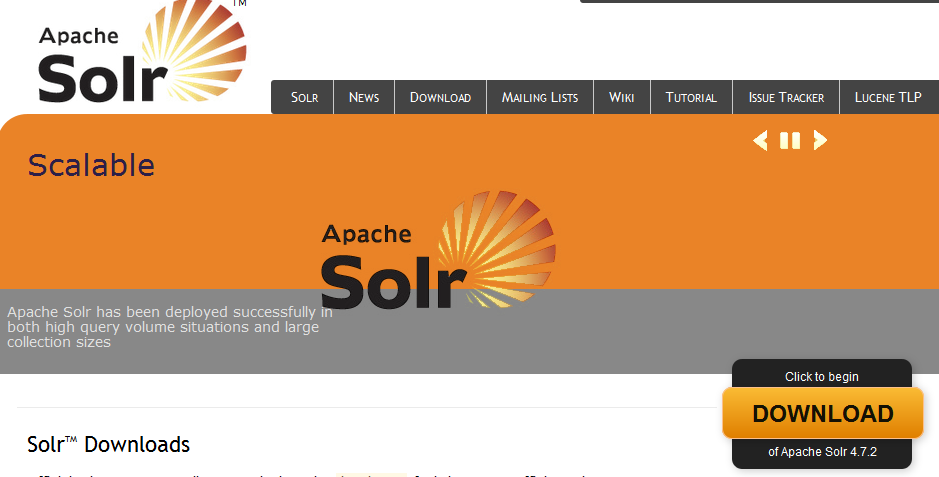
[11. Useful Notes 22](#_Toc386192862)

[a. Changing solr instance for drupal view: 22](#_Toc386192863)

How to configure solr for drupal

# Install Solr

* 1. Download latest version of solr from [http://lucene.apache.org/solr/downloads.html](http://lucene.apache.org/solr/downloads.html.)

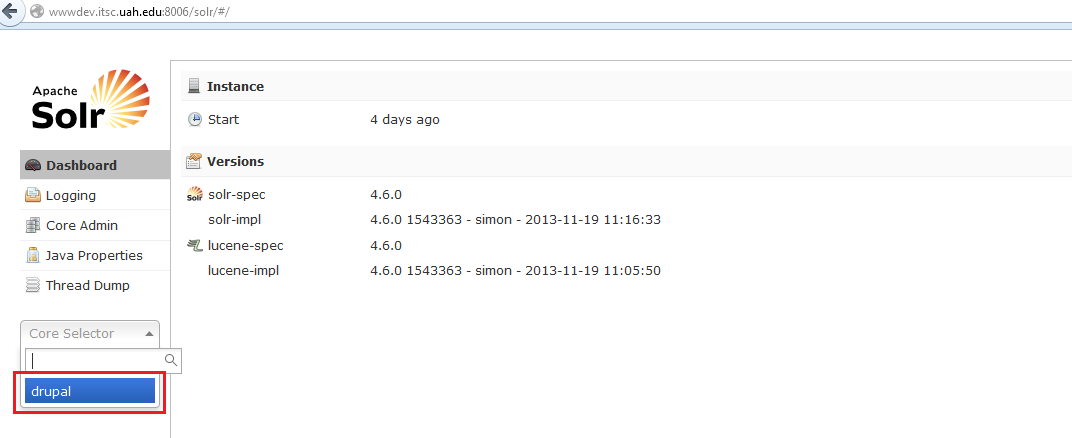


* 1. List of steps to install solr :
     1. Install java (OpenJDK min 1.6/ Tested version 1.7)
     2. Install the Tomcat server(Tested TOMCAT6):
     3. wget <http://supergsego.com/apache/lucene/solr/4.7.2/solr-4.7.2.zip>
     4. unzip solr-4.7.2.zip
     5. cd solr-4.7.2/example
     6. java -jar start.jar
     7. Check [http://localhost:8983/solr/](http://localhost:8983/solr/%20) in browser

# Configure Solr instance for Drupal

* 1. Download apache solr module for drupal :
     1. drush dl apachesolr
  2. Take backup of initial solr configuration :
     1. mv /home/user/solr-4.7.2/example/solr/collection1 collection1\_bkp
  3. Create a new folder for drupal instance:
     1. mkdir /home/ user /solr-4.7.2/example/solr/drupal
  4. Create a new folder for drupal configuration:
     1. mkdir /home/ user /solr-4.7.2/example/solr/drupal/conf
  5. Create a new folder for drupal data to be stored :
     1. mkdir /home/ user /solr-4.7.2/example/solr/drupal/data
  6. Copy the apache solr configuration provided by drupal apache solr module into the newly created folder :
     1. cp /home/user/apachesolr/solr-conf/solr-4.x/\* /home/ user /solr-4.7.2/example/solr/drupal/conf/
  7. Go to the example folder :
     1. cd /home/ user /solr-4.7.2/example
  8. Run the jar file to start solr service. It will now start solr with drupal configuration in the same port as before i.e. 8983:
     1. java -jar start.jar

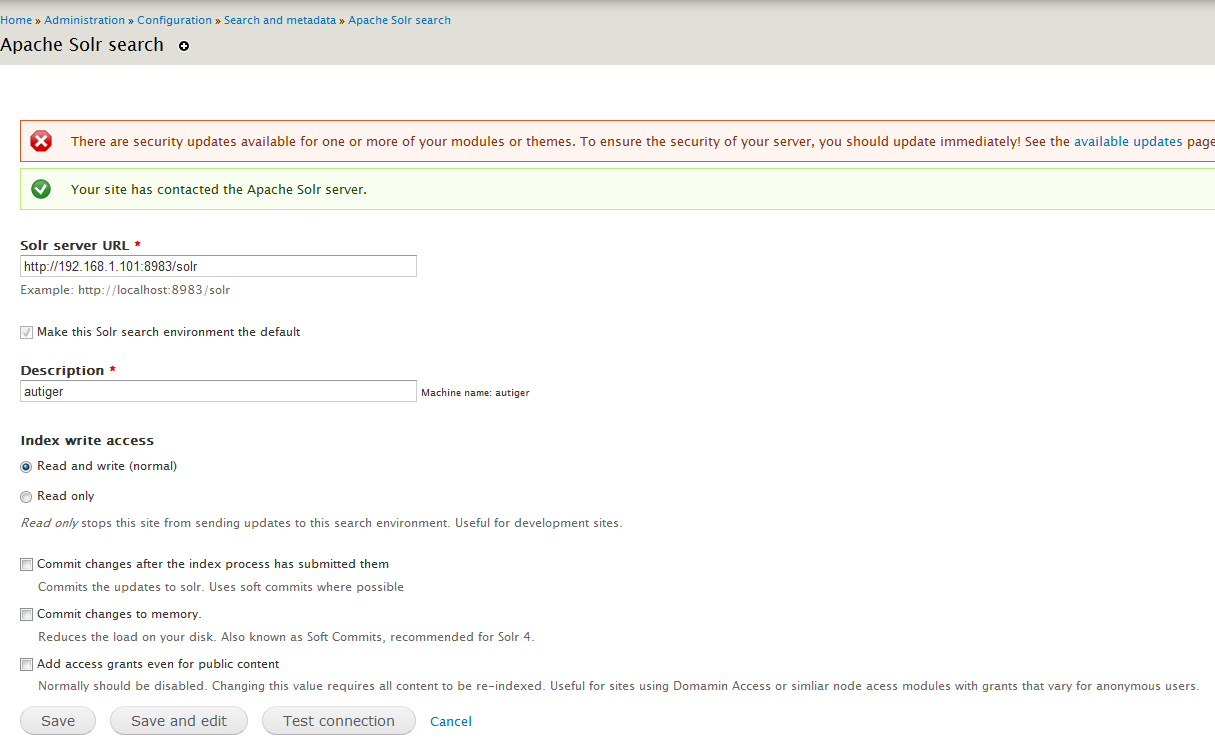
It should now show a new core called “drupal” as below:



# Drupal solr configurations

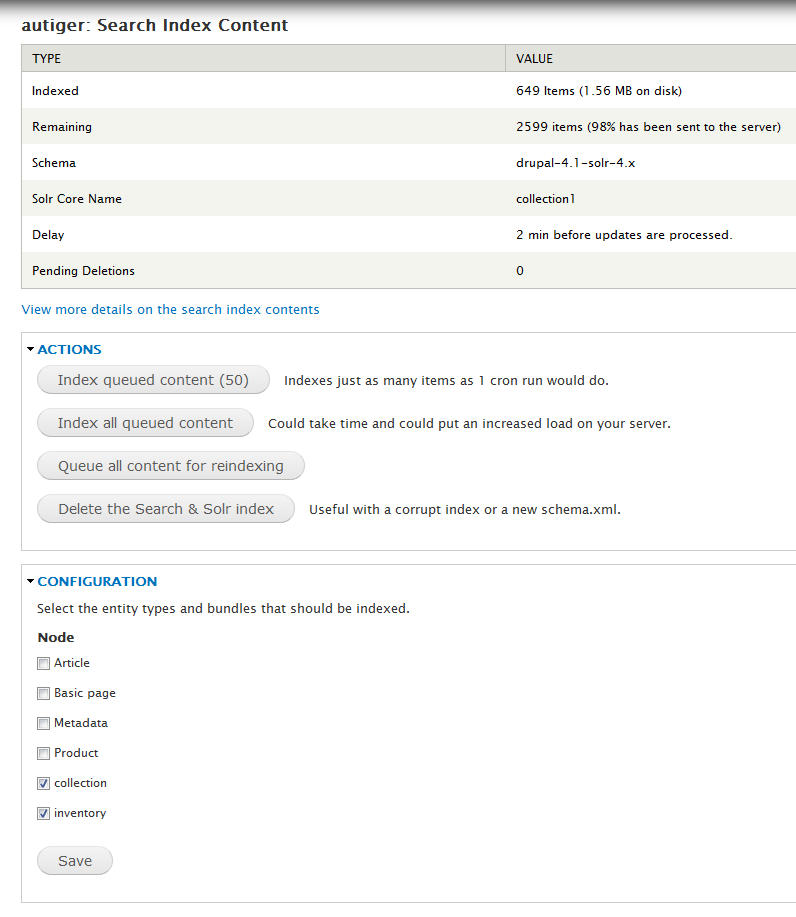
Install apache solr module for drupal (<https://drupal.org/project/apachesolr> ) and start solr instance. Configure solr settings for drupal from Configuration->Apache Solr Search -> settings.

Add Search Environment:



After adding solr location and name (Description), Test connection to verify if the solr instance is accessible from the drupal. If the drupal site will only be reading from solr indexes, then you may configure as read only, Otherwise Read and Write.

After saving the solr settings, go to “Default Index” tab :



Configuration shows list of content types to be indexed. Check the required ones. If the checked content types has any data, then it will appear as “Remaining”. There are 4 actions for solr indexing :

* 1. Index queued content (50): This is single click indexing for any 50 records from the remaining section of the checked content types.
  2. Index all queued content: This will index all remaining contents from the selected content types. This might take some time.
  3. Queue all content for re-indexing: If the content from the content types gets modified after indexing, then this action indexes only the modified ones. All other indexes are preserved.
  4. Delete the search and solr indexes: Deletes all indexes. Handy for schema change or fresh start.

# Indexing Field Collections

By default apache solr module does not index field collections. Install separate module for indexing field collection (<https://drupal.org/project/apachesolr_field_collection> ). **This module has not been maintained for a long time. Direct use of this module does not index field collections**. We need to modify module codes as below :

## Original Defective Code Snapshot :



## Corrected Snapshot

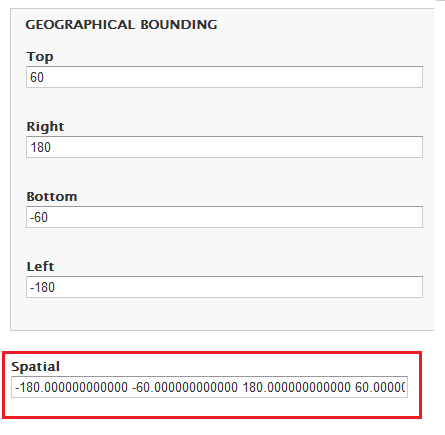
Add a line for looping all field collection fields

“foreach ($functions as $function) {“:



# Implementing Spatial Filters

* 1. Storing the spatial fields: First of all, the fields contacting spatial bounding box needs to be stored in a single field separated by **space**.



Spatial is the new field created to hold spatial data. Data needs to be stored starting from Left, Bottom, Right and Top. A lat-lon rectangle can be indexed with 4 numbers in minX minY maxX maxY order: Currently stored as sm\_field\_spatial[left bottom right top] (Ref: <http://wiki.apache.org/solr/SolrAdaptersForLuceneSpatial4> )

Below is the script I used for updating spatial field:

*<?php*

*//Script to alter collection content type for adding another field*

*//Drupal configuration*

*$\_SERVER['HTTP\_HOST'] = 'autiger.itsc.uah.edu';*

*$\_SERVER['REMOTE\_ADDR'] = '192.168.1.101';*

*$DRUPAL7\_ROOT = "/web/autiger.itsc.uah.edu/html/sod";*

*define('DRUPAL\_ROOT', $DRUPAL7\_ROOT);*

*chdir($DRUPAL7\_ROOT);*

*require\_once "./includes/bootstrap.inc";*

*drupal\_bootstrap(DRUPAL\_BOOTSTRAP\_FULL);*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*$param = array(*

*'type' => 'collection'*

*);*

*$type = "collection";*

*$nodes = node\_load\_multiple(array(), array('type' => $type));*

*$count =1;*

*foreach($nodes as $dataset){*

*echo 'Dataset '.$count."\n";*

*echo $dataset->nid;*

*$dataset->field\_spatial[und][0][value] = $dataset->field\_geographical\_bounding[und][0][left]." ".$dataset->field\_geographical\_bounding[und][0][bottom]." ".$dataset->field\_geographical\_bounding[und][0][right]." ".$dataset->field\_geographical\_bounding[und][0][top];*

*node\_save($dataset);*

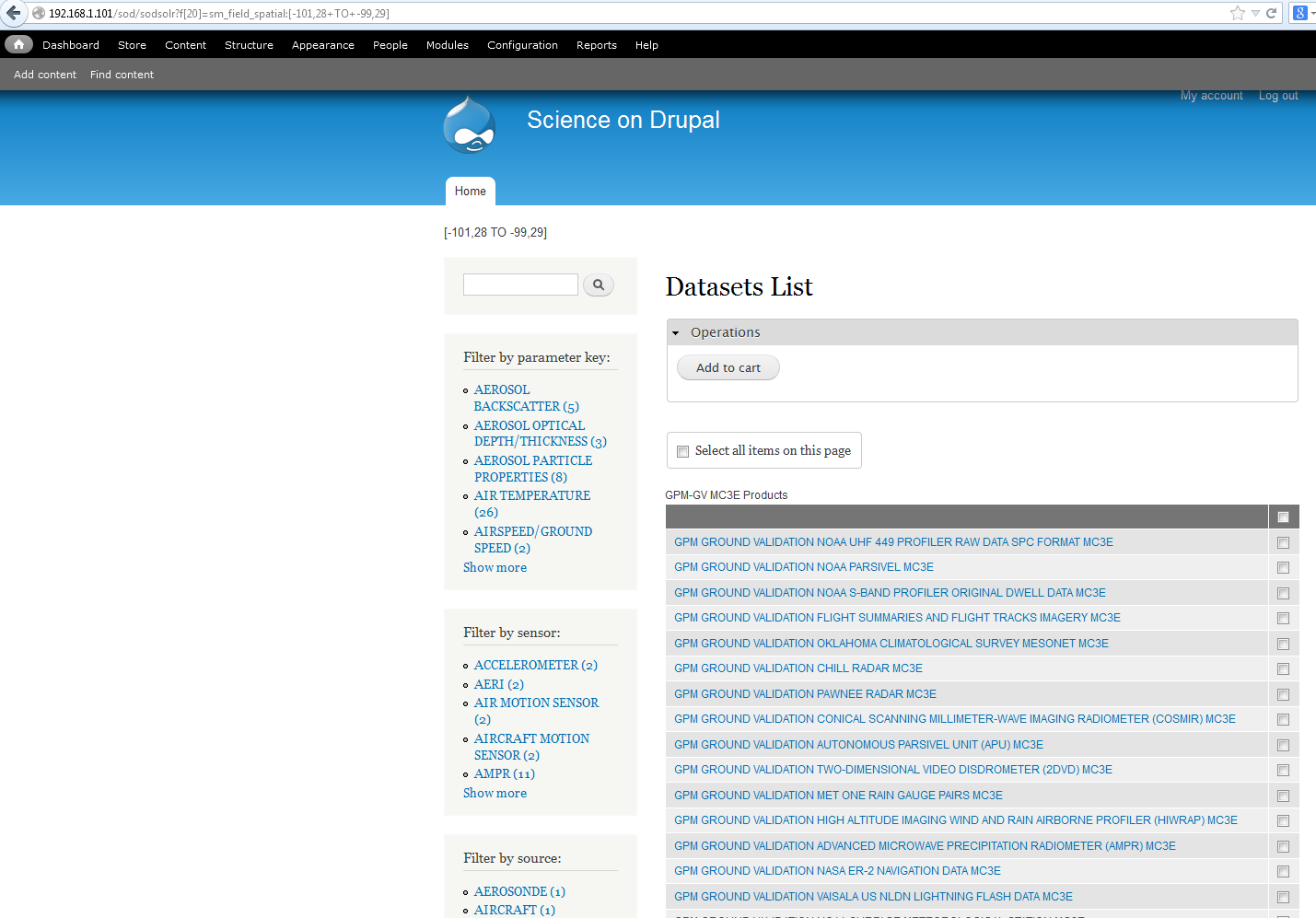
*$count ++;*

*echo "\n";*

*}*

* 1. Enable facets for the field sm\_field\_spatial. This makes sm\_field\_spatial field available for querying in the solr view itself. Since we don’t want to have facets for sm\_field\_spatial, we may continue to disable block for the spatial field.
  2. Create Drupal block : A drupal block needs to be created with 4 input textboxes using custom module.
  3. Input Parameters: On Submit code for the block should append parameters to the existing url as below:
     1. New URL to redirect = existing url + "f[20]=sm\_field\_spatial:["+west+","+south+"+TO+"+east+","+north+"]".
     2. Here west, south, north and east are 4 textbox input parameters and f[20] is the 20th argument for the solr query. Number 20 is chosen arbitrarily. solr adjusts argument number afterwards.
     3. Solr query for drupal view looks like below (Ref : <http://wiki.apache.org/solr/SpatialSearch> ) :

[http://192.168.1.101/sod/solr?f[0]=sm\_field\_spatial:[-101,28%20TO%20-99,29](http://192.168.1.101/sod/solr?f%5b0%5d=sm_field_spatial:%5b-101,28%20TO%20-99,29)]



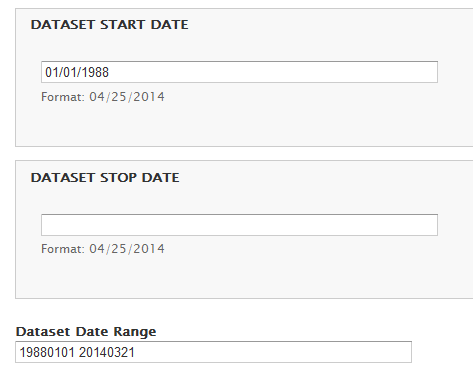
199 records displayed in test site out of 249 records with this filter. Verifying against solr instance :

[http://192.168.1.101:8983/solr/collection1/select?q=\*%3A\*&fq=sm\_field\_spatial%3A%5B-101%2C28+TO+-99%2C29%5D&wt=json&indent=true](http://192.168.1.101:8983/solr/collection1/select?q=*%3A*&fq=sm_field_spatial%3A%5B-101%2C28+TO+-99%2C29%5D&wt=json&indent=true)

# Implementing Temporal Filters:

* 1. Storing Temporal Fields :

Create a new field which stores temporal date range in space separated form. If start date or end date is null, then substitute it with appropriate coverage start and end date. Below is the snapshot:



Here Dataset Date Range is the new field created from dataset start and end date. Since end date is null, 20140321 is used as the end date. Dates field are in YYYYMMDD format. Below is the PHP script I used for this update:

*<?php*

*//Script to alter collection content type for adding another field*

*//Drupal configuration*

*$\_SERVER['HTTP\_HOST'] = 'autiger.itsc.uah.edu';*

*$\_SERVER['REMOTE\_ADDR'] = '192.168.1.101';*

*$DRUPAL7\_ROOT = "/web/autiger.itsc.uah.edu/html/sod";*

*define('DRUPAL\_ROOT', $DRUPAL7\_ROOT);*

*chdir($DRUPAL7\_ROOT);*

*require\_once "./includes/bootstrap.inc";*

*drupal\_bootstrap(DRUPAL\_BOOTSTRAP\_FULL);*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*$param = array(*

*'type' => 'collection'*

*);*

*$type = "collection";*

*$nodes = node\_load\_multiple(array(), array('type' => $type));*

*$count =1;*

*foreach($nodes as $dataset){*

*echo 'Dataset '.$count."\n";*

*//taking start date and converting it to YYYYMMDD*

*$start\_date\_array = explode('-', $dataset->field\_dataset\_start\_date[und][0][value]);*

*$day = explode(' ',$start\_date\_array[2]);*

*$start\_date = $start\_date\_array[0].$start\_date\_array[1].$day[0];*

*//taking end date and converting it to YYYYMMDD, if end date is null take today's date*

*if(!$dataset->field\_dataset\_end\_date[und][0][value]) // null value*

*$end\_date\_array = explode('-', date('Y-m-d'));*

*else*

*$end\_date\_array = explode('-', $dataset->field\_dataset\_end\_date[und][0][value]);*

*$day = explode(' ',$end\_date\_array[2]);*

*$end\_date = $end\_date\_array[0].$end\_date\_array[1].$day[0];*

*//updating date range field with startdate enddate*

*$date\_range = $start\_date." ".$end\_date;*

*$dataset->field\_dataset\_date\_range[und][0][value] = $date\_range;*

*node\_save($dataset);*

*$count ++;*

*}*

* 1. Enable facets for the field sm\_field\_dataset\_date\_range. This makes sm\_field\_dataset\_date\_range field available for querying in the solr view itself. Since we don’t want to have facets for sm\_field\_dataset\_date\_range, we may continue to disable block for the temporal field.
  2. Create Drupal block : A drupal block needs to be created with 2 input textboxes for calendar date picking using custom module.
  3. Input Parameters :

neg\_infinity = '2000-01-01T23:59:59.999Z'; var infinity = '2014-04-29T23:59:59.999Z'; New URL to redirect = existing url + “f[10]=dm\_field\_dataset\_stop\_date:["+start\_date+"+TO+"+infinity+"]&f[11]=dm\_field\_dataset\_start\_date:["+neg\_infinity+"+TO+"+end\_date+"]";

For temporal rectangle **Contains, Overlaps or Within** the range, use query as below:

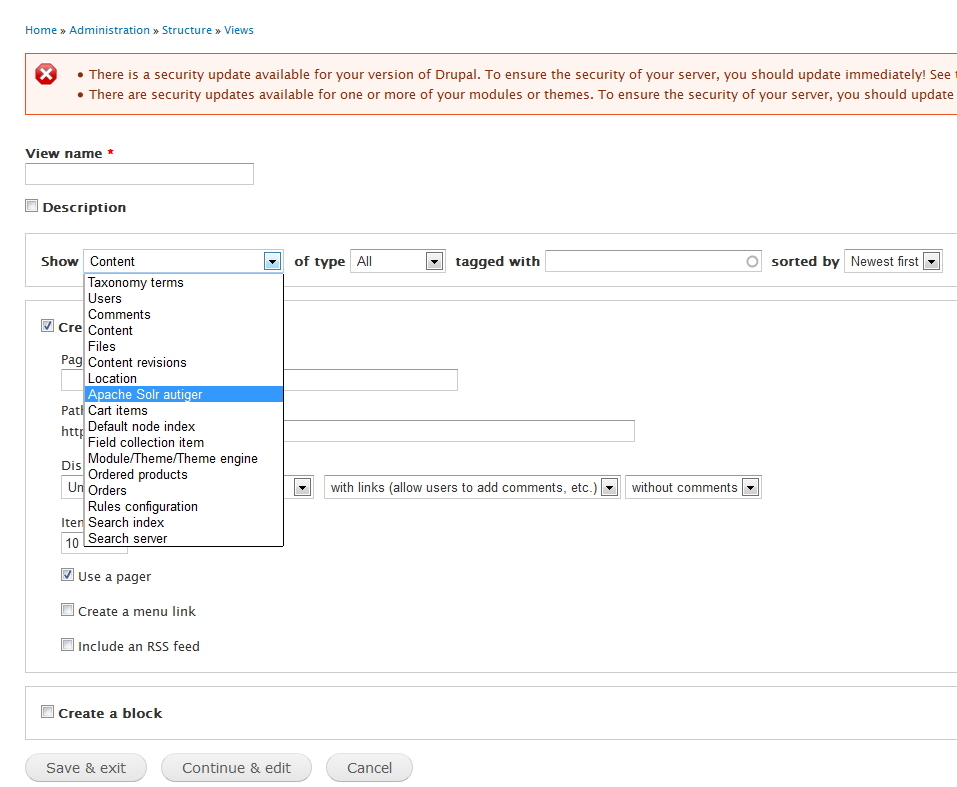
* Document Durration *Contains* Query Durration : q=fieldX:"Intersects(-∞ end start ∞)"
* Document Durration *Overlaps* Query Durration : q=fieldX:"Intersects(-∞ start end ∞)"
* Document Durration *Within* Query Durration : q=fieldX:"Intersects(start -∞ ∞ end)"

[Ref : https://people.apache.org/~hossman/spatial-for-non-spatial-meetup-20130117/]

# Creating Solr View

Requirement: Apache Solr Views (<https://drupal.org/project/apachesolr_views> ).

After installing above module for creating apache solr views, the apache solr instance appears while creating new view. Select the solr instance shown as below :



**Note : Once the apache solr instance is selected, view is tied to the solr instance. If the name of the solr instance changes, this view will break.**

This view will now work as normal drupal views except the list of fields are retrieved from the solr instance instead of drupal.

Issue faced while migrating solr views and resolution:

Sometimes the list of available solr fields was not showing the list of all fields. It was filtering some of the fields. This issue was suspected to be a cache issue, as solr query first looks for solr cache and if not found goes to solr instance to retrieve field names. It was resolved forcing the query to look for solr instance. Function setLuke from sites/all/modules/apachesolr/Drupal\_Apache\_Solr\_Service.php was modified as below :

***protected function setLuke****($num\_terms = 0) {*

*//custom ITSC comment so that it deliberately fetches it everytime*

*//if (empty($this->luke[$num\_terms])) {*

*$params = array(*

*'numTerms' => "$num\_terms",*

*'wt' => 'json',*

*'json.nl' => self::NAMED\_LIST\_FORMAT,*

*);*

*$url = $this->\_constructUrl(self::LUKE\_SERVLET, $params);*

*if ($this->env\_id) {*

*$cid = $this->env\_id . ":luke:" . drupal\_hash\_base64($url);*

*$cache = cache\_get($cid, 'cache\_apachesolr');*

*if (isset($cache->data)) {*

*$this->luke = $cache->data;*

*}*

*}*

*// }*

*// Second pass to populate the cache if necessary.*

*// if (empty($this->luke[$num\_terms])) {*

*$this->luke[$num\_terms] = $this->\_sendRawGet($url);*

*if ($this->env\_id) {*

*cache\_set($cid, $this->luke, 'cache\_apachesolr');*

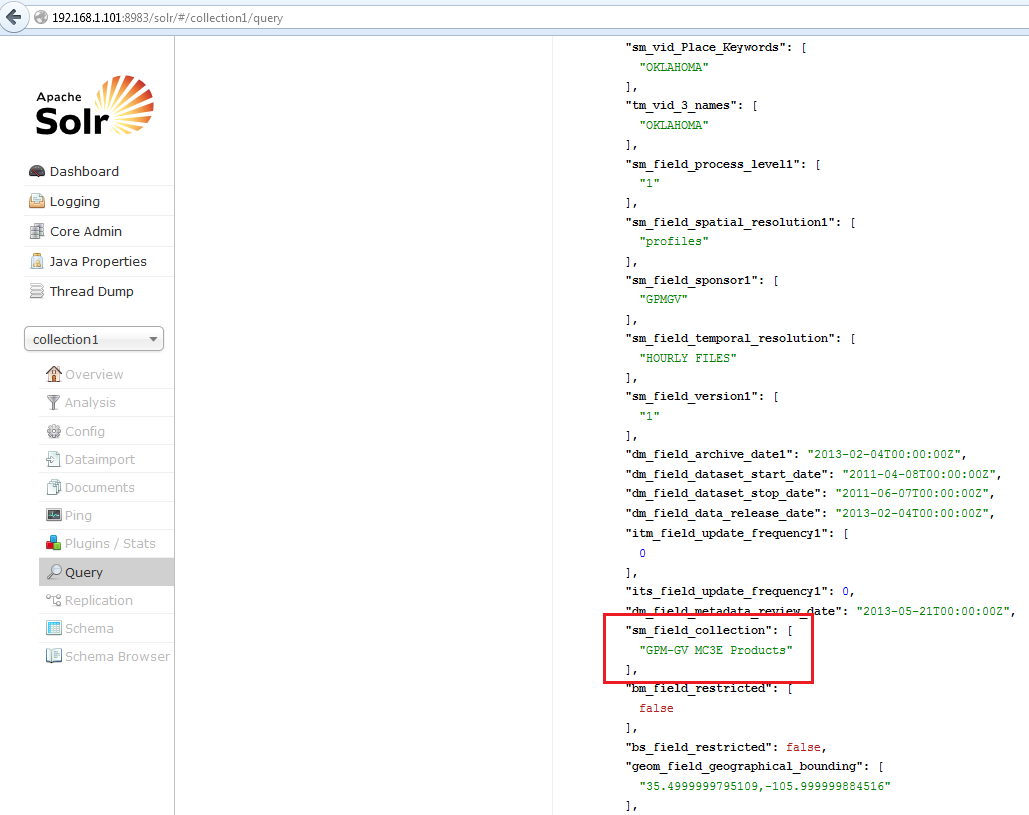
*}*

*// }*

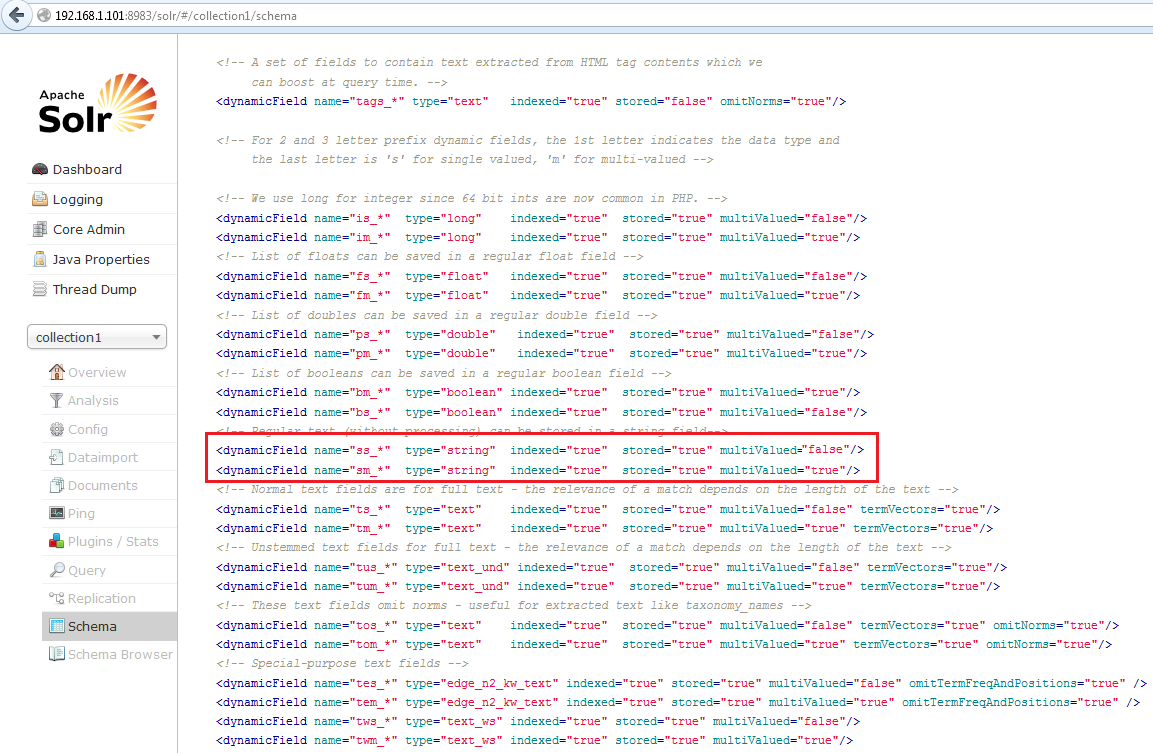
*}*

# Sorting solr view and modifying solr schema

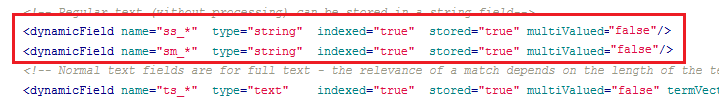
Any field can be used as a sorting field through “SORT CRITERIA” of the view or table view settings in case for tabular display of the view results. But the sorting field needs to be “single value field”. This criterion is specified in drupal solr schema which solr instance uses. For example:



If field\_collection needs to be used for sorting, it is stored as “sm\_field\_collection”. Here sm means string and multivalued. Therefore currently this field cannot be used for sorting. Schema would look something like below:



Therefore we have 2 options: either store field using single value field then it would be something like “ss\_field\_collection” or change the schema multivalued type for “sm\_field\_collection”. We tested by changing the schema type to single value and restarting solr service using the modified configuration as below :

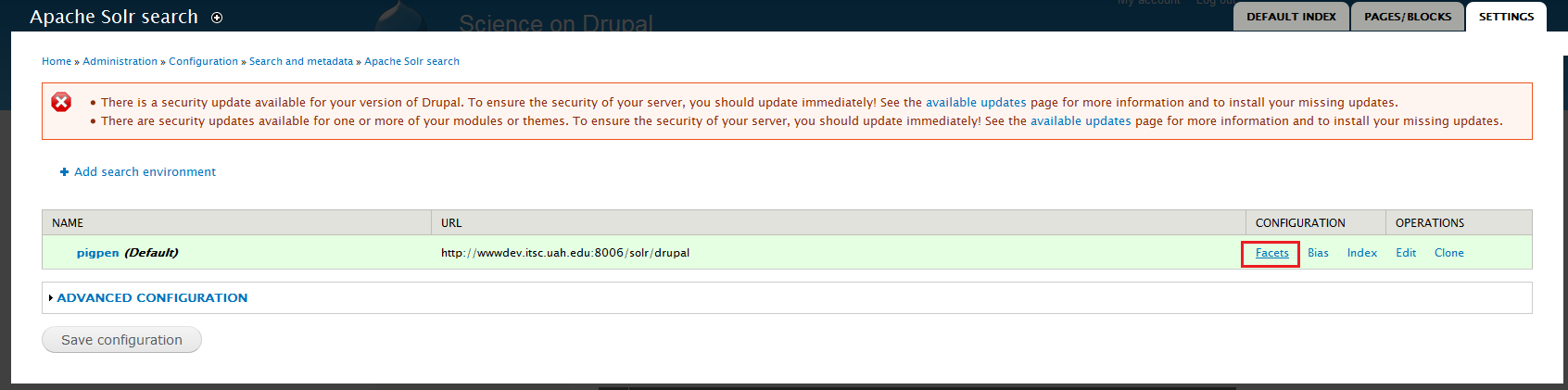


# Creating Facets

Requirements : Facet API (<https://drupal.org/project/facetapi> ).

Optional : Facetapi Slider (<https://drupal.org/project/facetapi_slider> ) for slider facets.

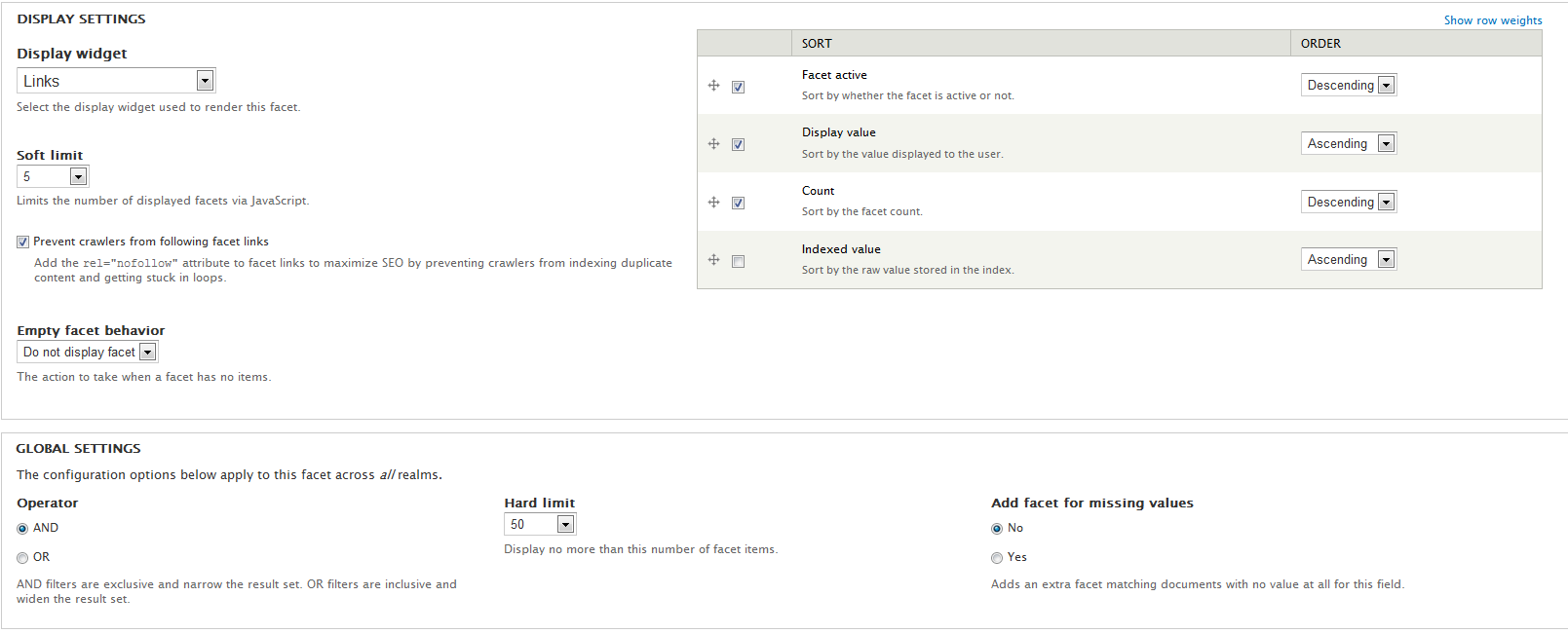
Facet configuration setting is available in the same apachesolr setting “admin/config/search/apachesolr/settings” of the default solr instance.



In facets, you can see list of all fields available for facets from the selected content type.

## Facet Configuration

Configure display for each facet will look like below:



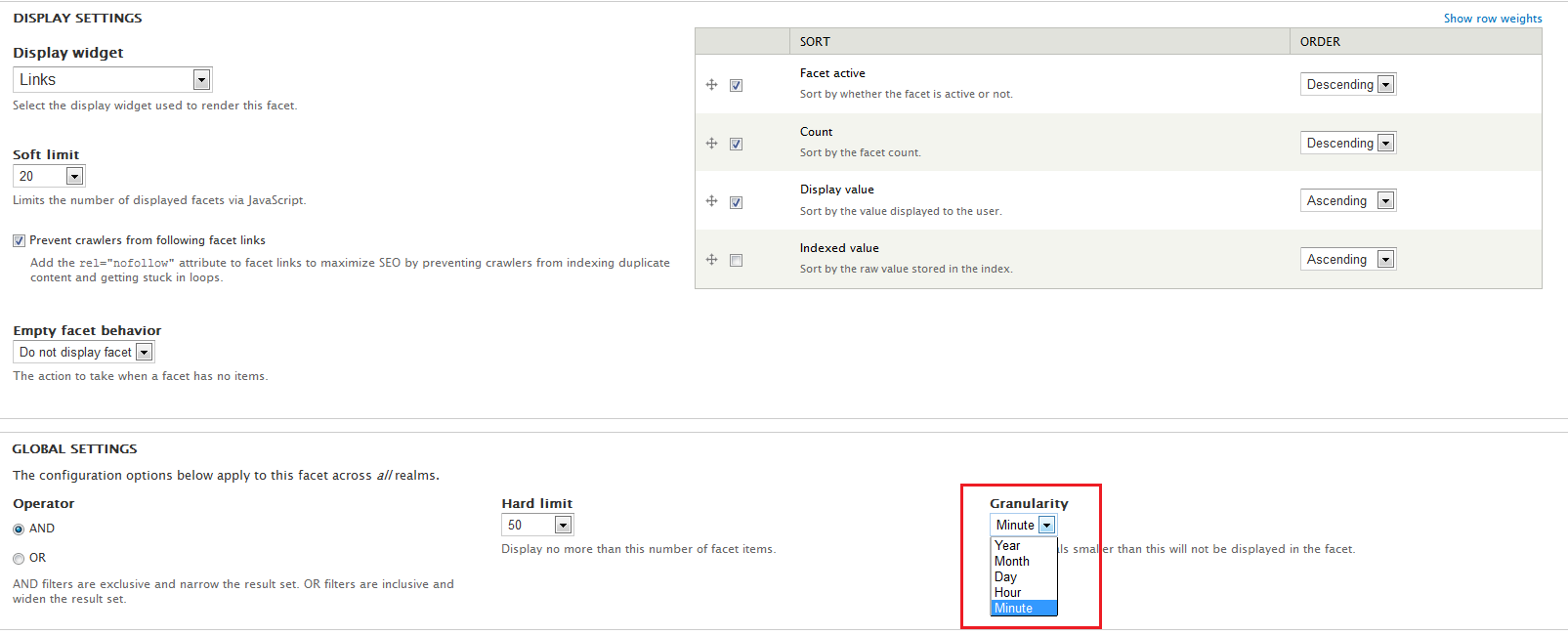
Mostly used configurations:

**Soft Limit** : To limit how many facet items to display with a show more option.

**Sort** : Sorting can be done alphabetically or based on facet count.

## Date time facets

For drupal content type fields classified as data type, a special date time granularity facet option is available if solr also stores the field as date time.



**Note : Solr facets are only displayed in solr views**.

After enabling facets from solr configuration, block of facets become available as regular blocks. Therefore from Structure-> Blocks we can configure these facets block same as any other blocks.

# Useful Drush Commands for solr indexing

* 1. drush solr-delete-index = :deletes all contents from the index.
  2. drush solr-delete-index node:article : Delete all content of the article content type
  3. drush solr-index : reindexes all marked contents. Accepts further argument for environment id and number of documents to index.

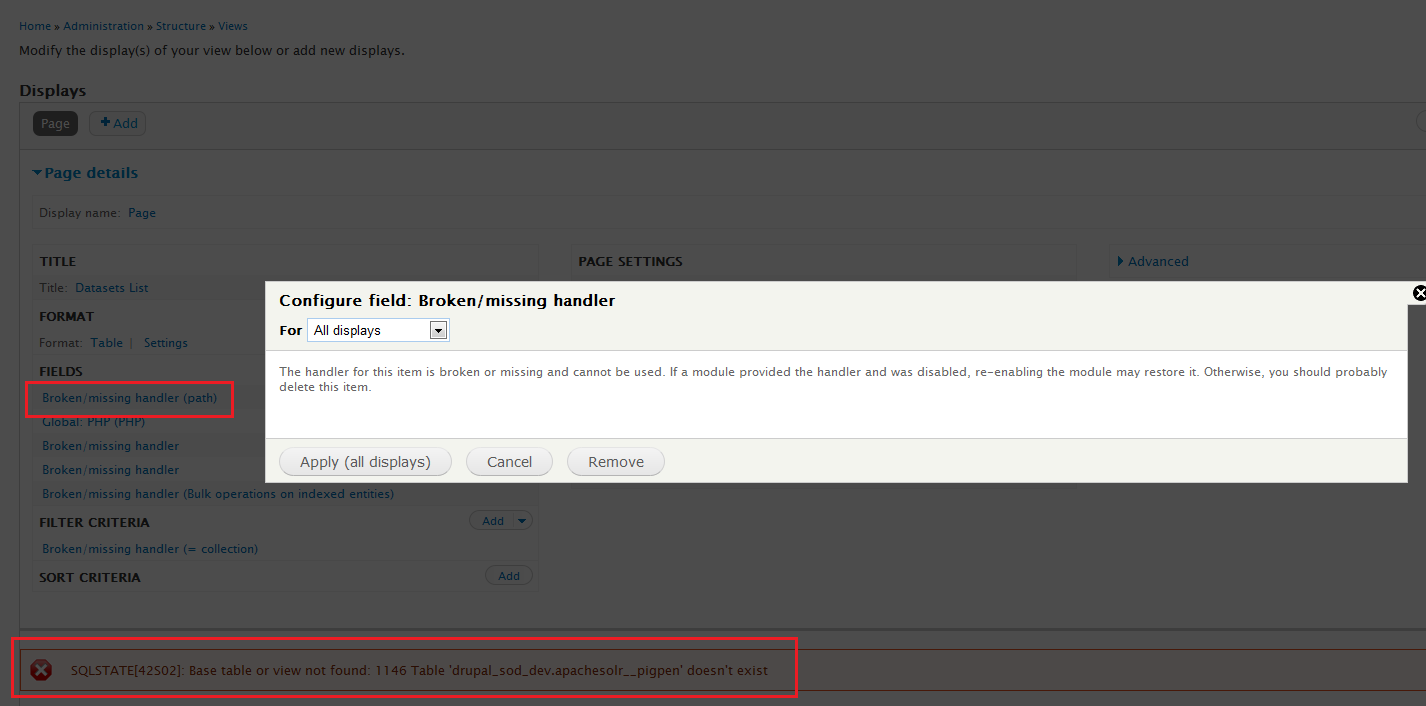
Full reference of drush commands for solr indexing : <http://www.drushcommands.com/drush-5x/apachesolr>

Tested version of drush is 5.9

# Useful Notes

## Changing solr instance for drupal view:

Drupal solr views query retrieve results based on the solr instance name. If the name of the solr instance changes views will be broken as below:



Therefore, it is advised to keep the name of the instance same even if the solr url changes.

