

 Consiglio Nazionale delle Ricerche

CERIS ISTITUTO DI RICERCA SULL'IMPRESA E LO SVILUPPO

Ottobre

2013

Rapporto tecnico N.45



STEP BY STEP INSTALLATION GUIDE OF A DIGITAL PRESERVATION INFRASTRUCTURE PART 2

Giancarlo Birello, Ivano Fucile,
Valter Giovanetti, Anna Perin



Consiglio Nazionale delle Ricerche



Istituto di Ricerche sull'Impresa e Lo Sviluppo

RAPPORTO TECNICO CNR-CERIS
Anno 8, N° 45; Ottobre 2013

Direttore Responsabile
Secondo Rolfo

Direzione e Redazione
CNR-Ceris
Istituto di Ricerca sull'Impresa e lo Sviluppo
Via Real Collegio, 30
10024 Moncalieri (Torino), Italy
Tel. +39 011 6824.911
Fax +39 011 6824.966
segreteria@ceris.cnr.it
<http://www.ceris.cnr.it>

Sede di Roma
Via dei Taurini, 19
00185 Roma, Italy
Tel. 06 49937810
Fax 06 49937884

Sede di Milano
Via Bassini, 15
20121 Milano, Italy
tel. 02 23699501
Fax 02 23699530

Segreteria di redazione
Enrico Viarisio
e.viarisio@ceris.cnr.it



Copyright © Ottobre 2013 by CNR-Ceris

STEP BY STEP INSTALLATION GUIDE OF A DIGITAL PRESERVATION INFRASTRUCTURE PART 2

*Giancarlo Birello**, *Ivano Fucile* *Valter Giovanetti*
(CNR-Ceris, IT Office)

Anna Perin
(CNR-Ceris, Library)

CNR-Ceris
IT Office
Strada delle Cacce, 73
10135 Torino – Italy
Phone: +39 011 3977533/4/5

CNR-Ceris
Library
Via Real Collegio, 30
10024 Moncalieri (Torino) – Italy
Phone: +39 011 6824928

* Corresponding author: g.birello@ceris.cnr.it

ABSTRACT: The Ceris-CNR project of digital preservation infrastructure has been committed by Bess (Social Science Electronic Library of Piemonte) for years 2011-2012 and confirmed for year 2013 sponsored by Compagnia di San Paolo of Turin.

Ceris-CNR role is to handle all the post-scan of the digitalization, for this purpose it has deployed the software and server platforms of the repository and also the web portal for the presentation, research and consulting. This report is the second part of step by step guide to build the digital archive infrastructure.

KEY WORDS: open-source, islandora, repository, digital archive, cms

Table of Contents

1 Introduction.....	5
2 Front-end server.....	6
2.1 Collapsed breadcrumb.....	6
2.2 Custom Theme.....	7
2.3 Custom Collection tabs.....	8
2.4 Custom search result.....	9
2.5 Custom Book and Page TABS.....	11
2.6 Oid-O-Matic connector.....	14
2.7 Custom Pager and Collection rendering.....	16
2.8 Multiple word search.....	17
2.9 Collection object for custom query and view.....	21
3 Repository.....	23
3.1 Add PICO metadata to OAI-PMH.....	23
3.2 Dissemination method for Low-Res JPEG from TIFF.....	26
4 Managing and ingesting scripts.....	28
4.1 Script to modify DC datastream dc:date.....	28
4.2 Generate BOOK files from PDF.....	29
4.3 Book ingesting script update.....	31
5 Webography.....	32
6 Appendix.....	33
6.1 Front-end Server.....	33
6.1.1 sites/all/modules/islandora-github/ObjectHelper.inc.....	33
6.1.2 sites/all/themes/digibess/digibess.css.....	44
6.1.3 sites/all/modules/islandora-github/CollectionClass.inc.....	46
6.1.4 sites/all/modules/islandora_solr_custom/theme/islandora_solr_custom.theme.inc.....	55
6.1.5 sites/all/modules/islandora_solr_custom/theme/islandora-solr-custom.tpl.php.....	58
6.1.6 sites/all/modules/islandora_solr_custom/css/islandora_solr_custom_tpl.css.....	59
6.1.7 sites/all/modules/islandora_solution_pack_book/tocnr_book.inc.....	60
6.1.8 sites/all/modules/islandora_solution_pack_book/xsl/tocnr_book_view.xsl.....	61
6.1.9 sites/all/modules/islandora_solution_pack_book/islandora_book.module.....	63
6.1.10 sites/all/modules/islandora-github/xsl/sparql_to_html.xsl.....	70
6.1.11 sites/all/modules/islandora_solr_search/IslandoraSolrResults.inc.....	74
6.1.12 sites/all/modules/islandora_solr_search/IslandoraSolrQueryProcessor.inc.....	80
6.1.13 sites/all/modules/islandora_solr_search/islandora_solr_search.module.....	83
6.2 Back-end server.....	88
6.2.1 ingpiubook.sh.....	88
6.2.2 ingbookepages.sh.....	88
6.2.3 templateBookMaster.xml.....	93
6.2.4 templatePageMaster.xml.....	94

1 Introduction

The CNR Ceris project of digital preservation infrastructure has been committed by Bess (Social Science Electronic Library of Piemonte) for years 2011-2012 and confirmed for year 2013.

Bess is a group of eighteen socioeconomic libraries in Piemonte (Italy) included Ceris library, they share a common specialization even if they are private foundations, research institutes, and university libraries that means different libraries in terms of size, parent institution, purpose, financial endowment, as well as collections.

One of the initiative promoted by Bess and sponsored by Compagnia di San Paolo of Turin, is the creation of a digital repository of sources of Piedmontese society and economy.

Bess has set up a digitalization laboratory, to be directly used by the members, for the conservation and preservation of collections included out of print and gray literature materials;

External partners are welcome to digitalize and share their archives, Istituto Gramsci for example has licensed "Sisifo" review as well as Centro Storico FIAT with "Illustrato Fiat" and "Blu Lancia" and Archivio Storico Lavazza with "Notizie Lavazza" that are now accessible through Bess archive. Other institutions have agree to share some collections yet.

The resulting repository is serving as a source of regional and economic information to the whole community.

CNR Ceris role is to handle all the post-scan of the digitalization and had to provide for the management of large volumes of data with the availability of space storage for the digitized works with characteristics of stability, versatility and dynamism. Currently (September, 2013) 416.751 pages are available in the repository.

CNR Ceris has deployed the software and server platforms of the repository, in a virtualized and redundant infrastructure and also take care of the design, development and management of the web portal (front-end) for the presentation, research and consulting data of the digitalized items. Moreover the repository is OAI-PMH compliant and is well harvested by the main National and International meta-repository .

Here listed some evidence of numbers, hardware and software used, most of the aspects will be analyzed in this paper or are available in the *"Step by step Installation Guide of a Digital Preservation Infrastructure."*, Rapporto Tecnico N.42, maggio 2012, CNR Ceris:

- files: pdf/a, high resolution tiff, txt file
- Metadata
- 2-node active/passive open-source cluster
- Kernel-based Virtual Machine (KVM) hypervisor
- repository: Fedora Commons
- harvesting OAI-PMH
- scripting for ingesting
- Custom models and datastreams
- front-end server: Drupal and Islandora
- Solr - search platform from the Apache Lucene project

The open-source community helped us to build on our project and this paper is at disposal of the open-source community, in particular we have to thank you Islandora Team with which we have exchanged many code pages.

2 Front-end server

The following customizations regard code modifications of Islandora 6.x modules in drupal front-end CMS as developed in the first part of this guide.

2.1 Collapsed breadcrumb

To limit title length in breadcrumb to avoid display overflow.

Edit Main breadcrumb (not Search breadcrumb) builder code:

```
nano -w sites/all/modules/islandora-github/ObjectHelper.inc [6.1.1]

...
function getBreadcrumbs($pid, &$breadcrumbs, $level=10) {
  module_load_include('inc', 'fedora_repository', 'api/fedora_utils');
  // Before executing the query, we have a base case of accessing the top-level collection
  global $base_url;
  if ($pid == variable_get('fedora_repository_pid', 'islandora:root')) {
    $breadcrumbs[] = l(t('Digital repository'), 'fedora/repository');
    $breadcrumbs[] = l(t('Home'), $base_url);
  }
  else {
    $query_string = 'select $parentObject $title $content from <#ri>
      where (<info:fedora/' . $pid . '> <dc:title> $title
      and $parentObject <fedora-model:hasModel> $content
      and (<info:fedora/' . $pid . '> <fedora-rels-ext:isMemberOfCollection> $parentObject
      or <info:fedora/' . $pid . '> <fedora-rels-ext:isMemberOf> $parentObject
      or <info:fedora/' . $pid . '> <fedora-rels-ext:isPartOf> $parentObject)
      and $parentObject <fedora-model:state> <info:fedora/fedora-system:def/model#Active>
      minus $content <mulgara:is> <info:fedora/fedora-system:FedoraObject-3.0>
      order by $parentObject';

    $query_string = htmlentities(urlencode($query_string));
    $url = variable_get('fedora_repository_url', 'http://localhost:8080/fedora/riearch');
    $url .= "?type=Tuples&flush=TRUE&format=CSV&limit=1&offset=0&lang=itql&stream=on&query=" . $query_string;

    $result = preg_split('/[\r\n]+/', do_curl($url));
    array_shift($result); // throw away first line
    $matches = str_getcsv(join("\n", $result));
    if ($matches !== FALSE) {
      $parent = preg_replace('/^info:fedora\/', '', $matches[0]);
      if ((strlen($parent) > 0) && ($level < 10)) {
        if (substr($matches[1], -11, 6) != '- page') {
          if (strlen($matches[1]) > 25) {
            $breadcrumbs[] = l(htmlentities(substr($matches[1], 0, 25)) . '...', 'fedora/repository/' . $pid);
          }
          else {
            $breadcrumbs[] = l($matches[1], 'fedora/repository/' . $pid);
          }
        }
      }
    }
    if ($parent == variable_get('fedora_repository_pid', 'islandora:root')) {
      $breadcrumbs[] = l(t('Digital repository'), 'fedora/repository');
      $breadcrumbs[] = l(t('Home'), $base_url);
    }
    elseif ($level > 0) {
      $this->getBreadcrumbs($parent, $breadcrumbs, $level - 1);
    }
  }
}
...

```

The following picture shows code result:



Picture 1: Collapsed breadcrumb

2.2 Custom Theme

Create a custom theme for drupal containing style definitions for standard and custom html tags.

- SSH login on drupal server
- Clone Garland theme

```
cd /usr/share/rootsitedir
mkdir sites/all/themes/bessparent
cp -R themes/garland/* sites/all/themes/bessparent/
rm -R sites/all/themes/bessparent/minnelli
```

- Configure parent theme

```
mv sites/all/themes/bessparent/garland.info
  sites/all/themes/bessparent/bessparent.info
nano -w sites/all/themes/bessparent/bessparent.info
name = DigiBessParent
description = Garland clone.
version = VERSION
core = 6.x
engine = phptemplate
stylesheets[all][] = style.css
stylesheets[print][] = print.css

; Information added by drupal.org packaging script on 2012-02-29
version = "6.25"
project = "drupal"
datestamp = "1330534547"
```

- Create child theme

```
mkdir sites/all/themes/digibess
nano -w sites/all/themes/digibess/digibess.info

name = DigiBess
description = DigiBess theme.
version = VERSION
core = 6.x
base theme = bessparent
stylesheets[all][] = digibess.css

; Information added by drupal.org packaging script on 2012-02-29
;version = "6.25"
;project = "drupal"
;datestamp = "1330534547"
```

- Create child template to override functions

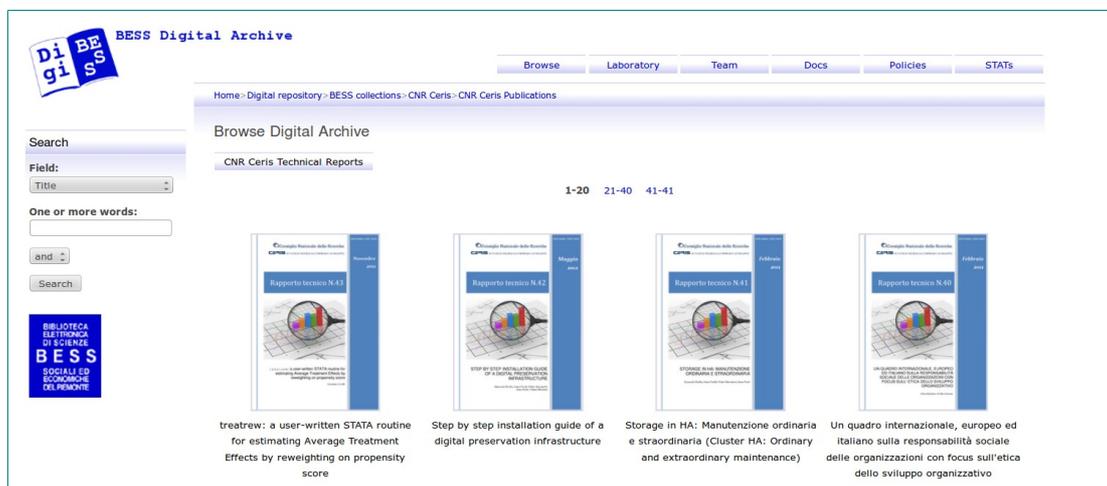
```
nano -w sites/all/themes/digibess/template.php

<?php
function digibess_breadcrumb($breadcrumb) {
  if (!empty($breadcrumb)) {
    return '<div class="breadcrumb">'. implode('>', $breadcrumb) . '</div>';
  }
}
```

- Create child CSS to override formatting

```
nano -w sites/all/themes/digibess/digibess.css [6.1.2]
```

- Copy modified *bg-content.png* in *sites/all/themes/digibess/images*
- Copy new *bg-tabs.png* in *sites/all/themes/digibess/images*
- Enable theme browsing to *siteURL/admin/build/themes*



Picture 2: DigiBESS theme example

2.3 Custom Collection tabs

Collection view tabs are defined in *CollectionClass.inc* file.

We modified this file to show custom title and a new tab for collection information page contained in INFO datastream.

```

nano -w sites/all/modules/islandora-github/CollectionClass.inc [6.1.3]

...
function showFieldSets($page_number) {
  module_load_include('inc', 'fedora_repository', 'api/fedora_item');
  global $base_url;
  $tabset = array();
  global $user;
  $objectHelper = new ObjectHelper();
  $item = new Fedora_Item($this->pid);
  $query = NULL;
  if ($item->exists() && array_key_exists('QUERY', $item->datastreams)) {
    $query = $item->get_datastream_dissemination('QUERY');
  }
  $results = $this->getRelatedItems($this->pid, $query);
  $colleinfo = NULL;
  if ($item->exists() && array_key_exists('INFO', $item->datastreams)) {
    $colleinfo = $item->get_datastream_dissemination('INFO');
  }
  $collection_items = $this->renderCollection($results, $this->pid, NULL, NULL, $page_number);
  $collection_item = new Fedora_Item($this->pid);
  // Check the form post to see if we are in the middle of an ingest operation.
  $show_ingest_tab = (!empty($_POST['form_id']) && $_POST['form_id'] == 'fedora_repository_ingest_form');
  $add_to_collection = $this->getIngestInterface();
  $view_selected = true;
  $coll_selected = false;
  $qstring = $_GET['q'];
  $qparts = explode('/', $qstring);
  $stail = end($qparts);
  if ($stail == 'info') {
    $view_selected = false;
    $coll_selected = true;
  }
  drupal_set_message();
  $tabset['1'] = array(
    '#type' => 'tabpanel',
    '#title' => $collection_item->objectProfile->objLabel,
    '#selected' => $view_selected,
    '#content' => $collection_items,
    '#weight' => '1'
  );
  if ($colleinfo != NULL) {
    $tabset['2'] = array(
      '#type' => 'tabpanel',
      '#title' => 'Collection info',
      '#selected' => $coll_selected,
      '#content' => $colleinfo,
      '#weight' => '2'
    );
  }
  return $tabset;
}
...

```

At repository level you have to add a datastream INFO to Collection object in text/html format with information about collection. Available header tags are h2 (without background) and h4 (with theme background), for example:

```

<html>
<head>
<meta content="text/html; charset=ISO-8859-1"
http-equiv="content-type">
<title>Lettera da Tecnocity</title>
</head>
<body>
<h2>Lettera da Tecnocity (1984 ? 1992)</h2>
<span style="font-weight: bold;">Direttore: Marcello Pacini</span><br>
<span style="font-weight: bold;">Un progetto della Fondazione Agnelli
per il rilancio dell'area metropolitana torinese</span><br>
<br>
<h4>LO SFONDO</h4>
<br>
Nel decennio tra gli anni ottanta e novanta la Fondazione Giovanni
Agnelli è impegnata in un ampio programma volto a studiare le
condizioni che rendono possibile e governano l'innovazione tecnologica
...

```

The following pictures show code result:



Picture 3: Custom collection Tabs



Picture 4: Collection INFO view

2.4 Custom search result

Search result view is customized with text highlighting if full-text search and some minor features here described.

- Speed up search and limit *dc.description* to 200 chars

```
nano -w sites/all/modules/islandora_solr_custom/theme/islandora_solr_custom.theme.inc [6.1.4]

...
function islandora_solr_custom_preprocess_islandora_solr_custom(&$variables)
{
  // base url
  global $base_url;
  $variables['base_url'] = $base_url;
  // include solr field order generated from solrconfig.xml
  // $variables['field_order'] = _islandora_solr_custom_solrconfig();
  // not sure if this is really needed, but I'll keep it in for now.
  // this variable might be more useful as the db values are parsed in the solrconfig order using the results of
  // _islandora_solr_custom_solrconfig().
  // get the solrconfig.xml and DB results. Don't check or fix any differences between solrconfig.xml and the DB
  $variables['db_values'] = _islandora_solr_custom_db_values($check = FALSE);
  ...
  if ($field == "PID")
  {
    $h1v = $hl->$value;
    foreach ($h1v as $singlef)
    {
      $linee = "";
      foreach ($singlef as $linea)
      {
        $linee = $linee . "... " . $linea . "... <br>";
      }
      $resultsArray['OCR.OCR']['value'] = $linee;
    }
  }
  if (($field == "dc.description") && ($value != ""))
  {

```

```

    $resultsArray[$field]['value'] = substr($value, 0, 200) . "... ";
}
// set the fields
$resultsArray[$field]['label'] = $db_value['label'];
$resultsArray[$field]['class'] = strtolower(preg_replace('/^[A-Za-z0-9]/', '-', $db_value['field'])); //
turn the field name in a string that is appropriate to use as a class name.
$resultsArray[$field]['exclude_label'] = $db_value['exclude_label'];
$resultsArray[$field]['markup'] = $db_value['markup'];
}
...
    
```

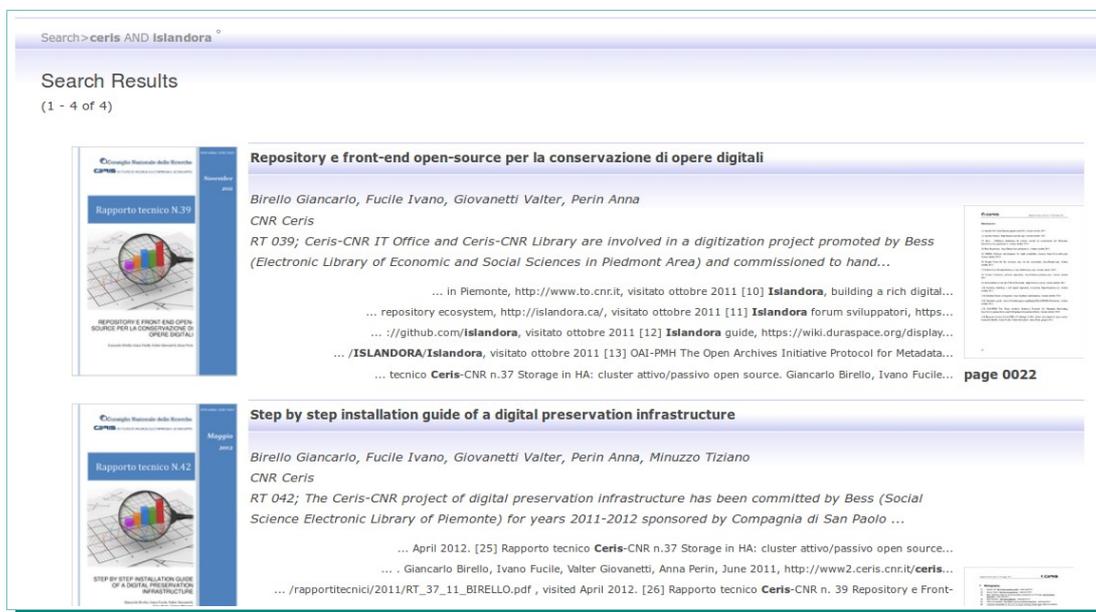
- Custom view of search result

We used Islandora Solr Custom module with a modified theme and css style sheet.

```

nano -w sites/all/modules/islandora_solr_custom/theme/islandora-solr-custom.tpl.php [6.1.5]
nano -w sites/all/modules/islandora_solr_custom/css/islandora_solr_custom_tpl.css [6.1.6]
    
```

Finally copy *bg-content.png* to *sites/all/modules/islandora_solr_custom/css/images* folder.



Picture 5: Full-text search result example

2.5 Custom Book and Page TABS

We made a lot of changes to book and page tabs and we fully customized main files rendering page and book view. In addition we added some code to Islandora Book module and Islandora basic module to reflect specific downloads for book and page.

- Custom code included into Islandora Book solution pack

```
nano -w sites/all/modules/islandora_solution_pack_book/tocnr_book.inc [6.1.7]
```

- Custom style sheet for rendering book and page view

```
nano -w sites/all/modules/islandora_solution_pack_book/xsl/tocnr_book_view.xsl [6.1.8]
```

- Index TAB rendering function added to Islandora solution pack book code

```
nano -w sites/all/modules/islandora_solution_pack_book/islandora_book.module [6.1.9]
```

```
...
function islandora_book_create_book_index($pid, $query = NULL)
{
  global $user;
  module_load_include('inc', 'fedora_repository', 'ObjectHelper');
  $path = drupal_get_path('module', 'islandora_book');
  $objectHelper = new ObjectHelper;
  $rels_xml = $objectHelper->getStream($pid, 'RELS-EXT');
  $simpleRELSxml = simplexml_load_string($rels_xml);
  $modello = $simpleRELSxml->xpath('//rdf:RDF/rdf:Description/fedora-model:hasModel/@rdf:resource');
  $parentcolle = $simpleRELSxml->xpath('//rdf:RDF/rdf:Description/fedora:isMemberOfCollection/@rdf:resource');
  $modello = $modello[0];
  $parentcolle = explode('/', $parentcolle[0]);
  $parentcolle = $parentcolle[1];
  if ($modello == 'info:fedora/islandora:bookCModel')
  {
    $index_html = $objectHelper->getStream($pid, 'INDEX');
    $ix_html = $objectHelper->getStream($pid, 'INDEX');
    $pidelbook = $pid;
    if (!$index_html)
    {
      drupal_set_message(t('Object does not exist.'), 'error');
      return '';
    }
  }
  else
  {
    $pidelbook = substr($pid, 0, -5);
    $index_html = $objectHelper->getStream($pidelbook, 'INDEX');
    $ix_html = $objectHelper->getStream($pidelbook, 'INDEX');
    if (!$index_html)
    {
      drupal_set_message(t('Object does not exist.'), 'error');
      return '';
    }
  }
  $str_html = "<table cellpadding=\"3\" cellspacing=\"2\" valign=\"top\" >";
  $str_html = $str_html . "<tr><td valign=\"top\">";
  $str_html = $str_html . "<img src=\"\" . base_path() . \"fedora/repository/\" . $pid . \"/TN\"></img>";
  $str_html = $str_html . "</td><td><div style=\"align:left\"><table cellpadding=\"3\" cellspacing=\"2\" width=\"90%\">";
  $str_html = $str_html . "<tbody><tr><td colspan=\"3\"><h3>Index</h3></td></tr>";
  $selenco = explode("|", $ix_html);
  $nvoci = sizeof($selenco);
  $voce = explode("|", $selenco[0]);
  $str_html = $str_html . "<tr><td><a href=\"\" . base_path() . \"fedora/repository/\" . $voce[0] . \"\"><h5> . $voce[1] . "</h5>";
  $str_html = $str_html . "</a></td><td></td></tr>";
  $i = 1;
  while ($i < $nvoci)
  {
    $voce = explode("|", $selenco[$i]);
    $str_html = $str_html . "<tr><td><a href=\"\" . base_path() . \"fedora/repository/\" . $voce[0] . \"\"> . $voce[1];
    $numpa = substr($voce[0], -4);
    $str_html = $str_html . "</a></td><td>page&nbsp;";
    $i++;
  }
  $str_html = $str_html . "</tbody></table></div></td></tr></table>";
  $strs[0] = $str_html;
  $colleinfo_html = $objectHelper->getStream($parentcolle, 'INFO');
  $strs[1] = $colleinfo_html;
  return $strs;
}
...

```

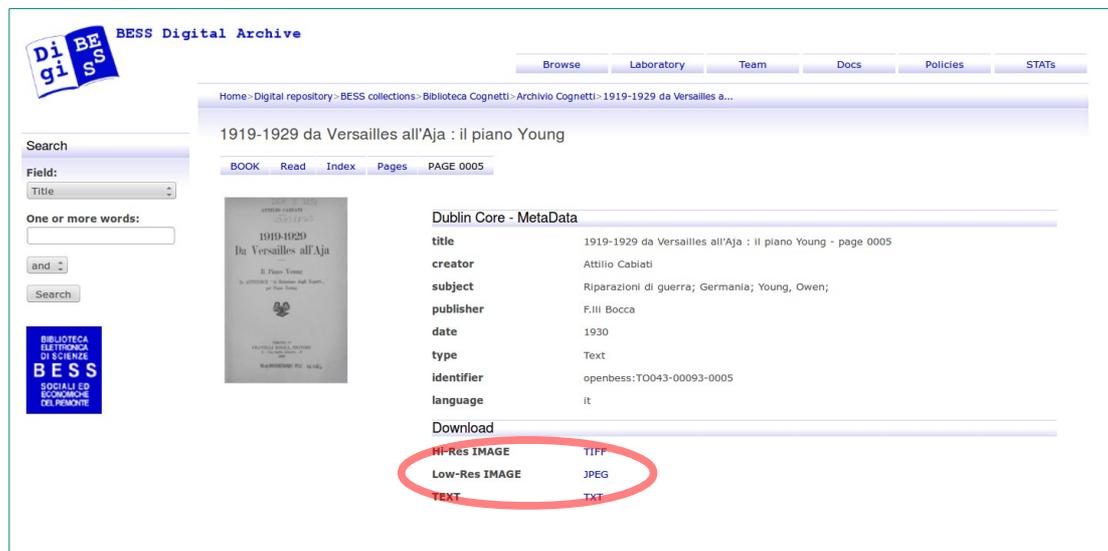
- Edit *ObjectHelper.inc* of Islandora main module to allow datastream download created by dissemination method

```

nano -w sites/all/modules/islandora-github/ObjectHelper.inc [6.1.1]
...
function makeObject($pid, $dsID, $asAttachment = FALSE, $label = NULL, $filePath=FALSE, $version=NULL, $forceSoap
= TRUE) {
...
    module_load_include('inc', 'fedora_repository', 'api/fedora_item');
    $item = new Fedora_Item($pid);
    if (isset($item->datastreams[$dsID])) {
        $mimeType = $item->datastreams[$dsID]['MIMETYPE'];
        if ($label == NULL) {
            $label = $item->datastreams[$dsID]['label'];
        }
    }
    else if ($dsID == "openbess:jpgldSdef") {
        $mimeType = "image/jpeg";
    }
    else {
        drupal_not_found();
        exit();
    }
    ...
    if (function_exists("curl_init")) {
        if (!isset($mimeType)) {
            $pid = variable_get('fedora_default_display_pid', 'demo:10');
            $dsID = variable_get('fedora_default_display_dsID', 'TN');
            $mimeType = 'image/jpeg';
        }
        if ($dsID == "openbess:jpgldSdef") {
            $url = variable_get('fedora_base_url', 'http://localhost:8080/fedora') . '/get/' . $pid .
            '/openbess:jpgldSdef/getJpgLD';
        }
        else {
            $url = variable_get('fedora_base_url', 'http://localhost:8080/fedora') . '/get/' . $pid . '/' . $dsID;
        }
        if ($version) {
            $url .= '/' . $version; //drupal_urlencode($version);
        }
        $ch = curl_init();
        $user_agent = "Mozilla/4.0 pp(compatible; MSIE 5.01; Windows NT 5.0)";
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
        curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, FALSE);
        curl_setopt($ch, CURLOPT_FAILONERROR, 1); // Fail on errors
        curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1); // allow redirects
        //curl_setopt($ch, CURLOPT_TIMEOUT, 15); // times out after 15s
        curl_setopt($ch, CURLOPT_USERAGENT, $user_agent);
        curl_setopt($ch, CURLOPT_URL, $url);
        curl_setopt($ch, CURLOPT_USERPWD, "$fedoraUser:$fedoraPass");
        // There seems to be a bug in Fedora 3.1's REST authentication, removing this line fixes the authorization
        denied error.
        // curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_ANY);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 0); // return into a variable
        curl_setopt($ch, CURLOPT_URL, $url);
        if ($filePath != FALSE) {
            $fp = fopen($filePath, 'w');
            curl_setopt($ch, CURLOPT_FILE, $fp);
            curl_setopt($ch, CURLOPT_HEADER, 0);
            curl_exec($ch);
            fclose($fp);
        }
        else {
            header("Content-type: $mimeType");
            if (($contentSize > 0) && ($dsID != "openbess:jpgldSdef")) {
                header("Content-length: $contentSize");
            }
            ...
        }
    }
    ...
}
...
    
```



Picture 6: Index Book view example and Book info Tab.



Picture 7: Download options in Page view

2.6 Oid-O-Matic connector

Oid-O-Matic is a web service catalog of Piedmont libraries Opac.

We included Oid-O-Matic harvesting in Book view tab.

- Add function for harvesting from Oid-O-Matic into Islandora solution pack book module

```
nano -w sites/all/modules/islandora_solution_pack_book/islandora_book.module [6.1.9]

...
function islandora_book_create_book_hardcopy($pid, $modello)
{
  $selelib = '';
  if ($modello == "info:fedora/islandora:bookCModel")
  {
    $url = "http://oidomatic.comperio.it/getinfo.xml?href=http%3A%2F%2Fwww.digibess.it%2Ffedora%2Frepository%2F"
    . $pid;
    $k = 0;
    $oidom = new DOMDocument();
    if (@$oidom->load($url) !== false)
    {
      if (is_object($root = $oidom->getElementsByTagName('objects')))
      {
        if (is_object($root->item(0)))
        {
          if (is_object($obj = $root->item(0)->getElementsByTagName('object')))
          {
            foreach ($obj as $obj)
            {
              $label = $obj->getElementsByTagName('label')->item(0)->nodeValue . "\n";
              $href = $obj->getElementsByTagName('href')->item(0)->nodeValue . "\n";
              if (($label !== '') && ($href !== ''))
              {
                $selelib .= "<p><a href='" . htmlentities($href, ENT_COMPAT, 'UTF-8') . "' target='_blank'>" .
                htmlentities($label, ENT_COMPAT, 'UTF-8') . "</a></p>";
                $k++;
              }
            }
          }
        }
      }
    }
  }
  return $selelib;
}
...
function islandora_book_create_book_view($pid, $query = NULL)
{
  ...
  $selelib = islandora_book_create_book_hardcopy($pid, $modello);
  //print_r($selelib);
  //inject into xsl stylesheet
  $proc->setParameter('', 'userID', $user->uid);
  $proc->setParameter('', 'objectsPage', base_path());
  $proc->setParameter('', 'pid', $pid);
  $proc->setParameter('', 'ingested', $ingested);
  $proc->setParameter('', 'modello', $modello);
  $proc->setParameter('', 'selelib', $selelib);
  $xsl = new DomDocument();
  $test = $xsl->load($path . '/xsl/tocnr_book_view.xsl');
  ...
}
```

- Add harvesting result to book view style sheet

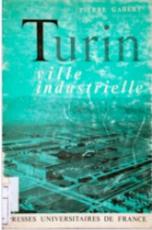
```
nano -w sites/all/modules/islandora_solution_pack_book/xsl/tocnr_book_view.xsl [6.1.8]

...
<xsl:if test="$MODELLO = 'info:fedora/islandora:bookCModel'">
  <xsl:if test="$SELELIB != ''">
    <tr>
      <td colspan="3">
        <h3/>
      </td>
    </tr>
    <tr>
      <td colspan="3">
        <h3>External link</h3>
      </td>
    </tr>
    <tr>
      <td valign="top">
        <strong>Hard copy</strong>
      </td>
      <td valign="top">
        <xsl:value-of disable-output-escaping="yes" select="$SELELIB"/>
      </td>
    </tr>
  </xsl:if>
</xsl:if>
...

```

Turin: ville industrielle : etude de geographie economique et humaine

[BOOK](#) [Read](#) [Index](#) [Pages](#) [PAGE 0001](#) [Book info](#)



Dublin Core - MetaData

title	Turin: ville industrielle : etude de geographie economique et humaine
creator	Pierre Gabert ; preface de Raoul Blanchard
subject	Torino; Fiat; Industria automobilistica; Sviluppo economico
publisher	Presse universitaires de France
date	1964
type	Text
identifrier	openbess:TO082-00017
identifrier	http://www.digibess.it/fedora/repository/openbess:TO082-00017
language	fr

Download

Full PDF [PDF](#)

External link

Hard copy

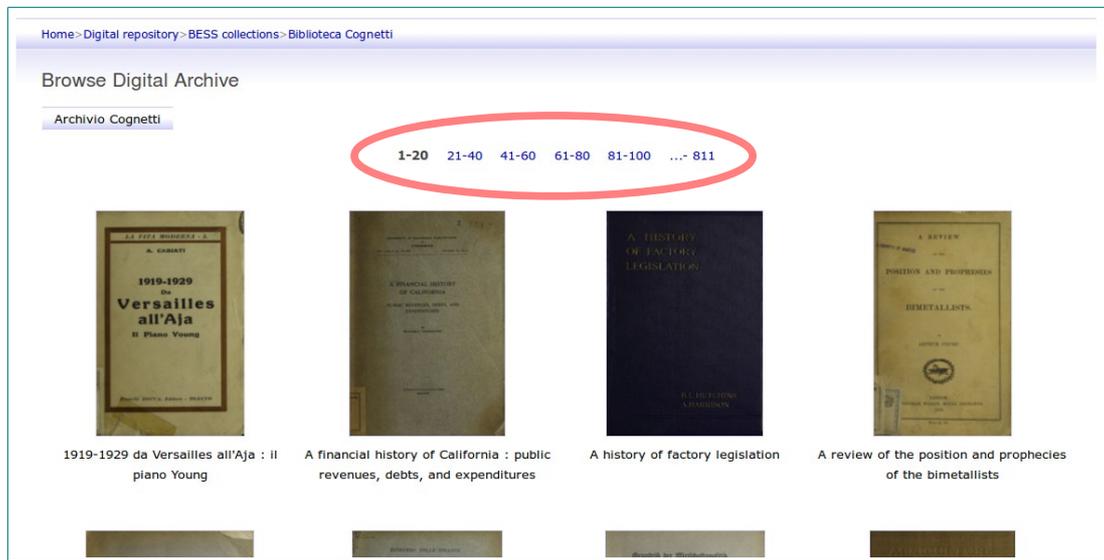
- [Librinlinea](#)
- [Biblioteche Civiche Torinesi](#)
- [Polo SBN dell'Università degli Studi di Torino](#)

Picture 8: Oid-o-Matic harvesting result view

2.7 Custom Pager and Collection rendering

We customized how collection objects are rendered and multiple page browsing.

```
nano -w sites/all/modules/islandora-github/xsl/sparql_to_html.xsl [6.1.10]
```



Picture 9: Collection view and custom pager

We can change the Islandora default ordering of collection elements editing QUERY datastream of collection. For example we set ordering by DC Description element (while default is by DC Title) with this ITQL query stored in QUERY datastream:

```
select $object $title $content $description from <#xi>
where ($object <dc:title> $title
and $object <dc:description> $description
and $object <fedora-model:hasModel> $content
and ($object <fedora-rels-ext:isMemberOfCollection> %parent_collection%
or $object <fedora-rels-ext:isMemberOf> %parent_collection%)
and $object <fedora-model:state> <info:fedora/fedora-system:def/model#Active>)
minus $content <mulgara:is> <info:fedora/fedora-system:FedoraObject-3.0>
order by $description
```

NOTE: %parent_collection% will be replaced with <info:fedora/' . \$pid. '> by Islandora php code.

2.8 Multiple word search

Extend basic function of Islandora Solr search module to multiple word search with AND/OR condition.

- Add new fields to Islandora Solr search form

```
nano -w      sites/all/modules/islandora_solr_search/IslandoraSolrResults.inc      [6.1.11]

...
function build_solr_search_form($repeat = NULL, $pathToSearchTerms = NULL, $query = NULL)
{
  $types = array();
  $terms = trim(variable_get('islandora_solr_searchterms', 'dc.title ~ Title,dc.subject ~ Subject'));
  $termsArray = preg_split('/[\n]/', $terms);
  foreach ($termsArray as $term)
  {
    $vals = split('~', $term);
    if (!$vals[1])
    {
      $vals[1] = $vals[0];
    }
    $types[trim($vals[0])] = trim($vals[1]);
  }
  $queryArray = NULL;
  if (isset($query))
  {
    $stermini = explode(":", $query);
    $andOrs[] = trim($stermini[2]);
    $queryArray = preg_split('/( OR | AND )/', $query, 0, PREG_SPLIT_NO_EMPTY | PREG_SPLIT_DELIM_CAPTURE);
    $queryArray = array_values($queryArray);
  }
  $andOrArray = array(
    'AND' => 'and',
    'OR' => 'or'
  );
  $form = array();
  if (!isset($repeat))
  {
    $repeat = variable_get('islandora_solr_search_block_repeat', '3');
  }
  $var0 = explode(':', $queryArray[0]);
  $form['search_type']['type'] = array(
    '#title' => 'Field',
    '#type' => 'select',
    '#options' => $types,
    '#default_value' => trim($var0[0])
  );
  $form['fedora_terms1'] = array(
    '#size' => '24',
    '#type' => 'textfield',
    '#title' => 'One or more words',
    '#default_value' => (count($var0) >= 2 ? restoreSlashes(trim($var0[1], '()')) : '')
  );
};
if ($repeat > 0 && $repeat < 9) //don't want less than 2 or more than 9
{
  for ($i = 1; $i < $repeat + 1; $i++)
  {
    $t = $i - 1;
    //      $j = $i - 2;
    $j = $i - 1;
    $andOrj = (isset($andOrs[$j]) ? $andOrs[$j] : 'AND');
    $field_and_term = explode(':', $queryArray[$t]);
    $form["andor$t"] = array(
      '#title' => '',
      '#type' => 'select',
      '#default_value' => $andOrj,
      '#options' => $andOrArray
    );
  }
}
$form['submit'] = array(
  '#type' => 'submit',
  '#value' => t('Search')
);
return $form;
}
...
function theme_solr_search_form($form)
{
  if (!isset($repeat))
  {
    $repeat = variable_get('islandora_solr_search_block_repeat', '3');
  }
  $output = drupal_render($form['search_type']['type']);
  $output .= drupal_render($form['fedora_terms1']);
  $output .= drupal_render($form['andor1']);
  $output .= drupal_render($form);
  return $output;
}
}
nano -w      sites/all/modules/islandora_solr_search/islandora_solr_search.module      [6.1.13]
```

```

...
function islandora_solr_search_block_form_submit($form, &$form_state)
{
    global $queryClass;
    islandora_solr_search_init();
    $type_id = trim($form_state['values']['type']);
    $repeat = variable_get('islandora_solr_search_block_repeat', '3');
    $fedora_terms = array();
    $types = array();
    $booleans = array();
    for ($fieldNum = 1; $fieldNum <= $repeat; $fieldNum++)
    {
        if ($form_state['values']["fedora_terms$fieldNum"])
        {
            $types[] = trim($form_state['values']["type$fieldNum"]);
            $fedora_terms[] = lesser_escape(trim($form_state['values']["fedora_terms$fieldNum"]));
        }
        $next = $fieldNum + 1;
        if ($form_state['values']["fedora_terms$next"] && $form_state['values']["fedora_terms$fieldNum"])
        {
            $booleans[] = trim($form_state['values']["andor$fieldNum"]);
        }
    }
    for ($iteration = 0; $iteration < count($fedora_terms); $iteration++)
    {
        $searchString .= $types[$iteration] . ":" . $fedora_terms[$iteration] . " " . $booleans[$iteration] . " ";
    }
    $searchString .= ":" . trim($form_state['values']["andor0"]);
    $searchString = trim($searchString);
    $searchString = htmlspecialchars(drupal_urlencode($searchString), ENT_QUOTES, 'utf-8', false);
    $searchString = str_replace('/', '~slash-', $searchString); //replace the slash so url doesn't break
    drupal_goto("islandora/solr/search/$searchString/-");
}
...

```

- Modify Solr query builder

```

nano -w sites/all/modules/islandora_solr_search/IslandoraSolrQueryProcessor.inc [6.1.12]
...
function buildQuery($query, $fq = null, $dismax = null)
{
    // Get pager variable
    $startPage = isset($_GET['page']) ? $_GET['page'] : 0;
    $this->solrLimit = variable_get('islandora_solr_search_num_of_results', 20);
    global $base_url, $facetArray; //used by facet block as well
    // -- cut here --
    // Get display profile
    $islandora_solr_primary_display = variable_get('islandora_solr_primary_display', 'default');
    $profiles = module_invoke_all("islandora_solr_primary_display");
    if (isset($profiles[$islandora_solr_primary_display]))
    {
        drupal_set_message("There is an error in the solr search configuration: the display profile is not found.",
        'error');
        return "";
    }
    $profile = $profiles[$islandora_solr_primary_display];
    // Include the file for the display profile
    require_once(drupal_get_path('module', $profile['module']) . '/' . $profile['file']);
    // Set display class and function vars
    $solrClass = $profile['class'];
    $solrFunction = $profile['function'];
    // -- end cut here --
    //fix the query as some characters will break the search : and / slash are examples
    $this->solrQuery = restoreSlashes($query);
    if (empty($this->solrQuery))
    {
        // $this->solrQuery = '%20'; //so we can allow empty queries to dismax
    }
    $facetArray = array();
    $facetFields = '';
    $rawFacetVals = variable_get("islandora_solr_search_block_facets", 'dc.subject,dc.type');
    $facetArray = islandora_build_substitution_list($rawFacetVals);
    $facetFields = implode(",", array_keys($facetArray));
    $facetLimit = variable_get('islandora_solr_search_block_facet_limit', '12');
    $facetMinCount = variable_get('islandora_solr_search_block_facet_min_count', '2');
    $requestHandler = variable_get("islandora_solr_search_block_request_handler", "standard");
    $this->solrParams = array(
        'facet' => 'true',
        'facet.mincount' => $facetMinCount,
        'facet.limit' => $facetLimit,
        'qt' => $requestHandler,
        'facet.field' => explode(',', $facetFields) //comma separated list configured in the block config
    );
    $debugQuery = (variable_get('islandora_solr_search_debug_mode', 0) ? "true" : null); //any val incl. 'false'
    is treated as true by Solr
    // $fq = preg_replace('/PID:.*\/', '', $fq);
    if ($fq != NULL && $fq != '-')
    {
        $fq = restoreSlashes($fq); //put the slash back
        $fqs = csv_explode(IslandoraSolrQueryProcessor::$facetSeparator, $fq, '"', true); //to filter by more than
        one facet we will separate them by~ for nowseparate them by~ for now
        $this->solrParams['fq'] = $fqs;
        $islandora_fq = replaceSlashes($fq); //remove the slash here as we will be using this in url's
    }
}

```

```

}
if (empty($islandora_fq))
{
  $islandora_fq = '-';
}
if ($dismax != NULL)
{
  $this->solrDefType = $dismax;
  $this->solrParams['defType'] = $dismax;
}
$this->solrStart = max(0, $startPage) * $this->solrLimit;
// The breadcrumb should go in the display class
$queryurl = "islandora/solr/search/" . replaceSlashes($this->solrQuery);
$breadcrumb_fq = $islandora_fq;
if (strcmp($islandora_fq, "-"))
{
  foreach (csv_explode(IslandoraSolrQueryProcessor::$facetSeparator, $islandora_fq, '', true) as $facet)
  {
    $field = substr($facet, 0, strpos($facet, ":"));
    if ($field != "rels.hasModel")
    {
      $value = restoreSlashes(trim(substr($facet, strpos($facet, ":") + 1, ""));
      $options = array(
        'html' => true
      );
      $options['attributes']['title'] = $facet;
      $crumbLink = $queryurl . "/" . $breadcrumb_fq . (empty($this->solrDefType) ? '' : '/' . $this->solrDefType);
      $cut_fq = $this->delete_filter($islandora_fq, $facet);
      $cutlink = $queryurl . "/" . $cut_fq . (empty($this->solrDefType) ? '' : '/' . $this->solrDefType);
      if (!strncmp("-", $facet, 1))
      {
        $options['attributes']['class'] = "strikethrough";
      }
      $breadcrumb[] = l($value, $crumbLink, $options) . " <span
class='islandora_solr_search_breadcrumb_super'>(" . l("x", $cutlink, array(
  'attributes' => array(
    'title' => "Remove " . $facet
  )
) . ")</span>";
      $breadcrumb_fq = $this->delete_filter($breadcrumb_fq, $facet);
    }
  }
}
if (!empty($this->solrQuery) && strcmp($this->solrQuery, ' '))
{
  $cutlink = "islandora/solr/search/ /" . $islandora_fq . "/dismax";
  $queryval = $this->solrQuery;
  $tokens = explode(" ", $queryval);
  foreach ($tokens as $token)
  {
    preg_match("/^.*:/", $token, $matches);
    $queryval = str_replace($matches[0], "", $queryval);
  }
  if ($queryval != '%252F')
  {
    $elencoparole = preg_replace('~\s+~', ' ', trim($queryval));
    $searchwords = explode(" ", $elencoparole);
    $elencoparole = "<strong>" . $searchwords[0] . "</strong>";
    $nparole = count($searchwords);
    if ($nparole > 2)
    {
      for ($i = 1; $i < ($nparole - 1); $i++)
      {
        $elencoparole .= " " . $searchwords[$nparole - 1] . " " . "<strong>" . $searchwords[$i] .
"</strong>";
      }
    }
    $elencoparole .= " <span class='islandora_solr_search_breadcrumb_super'></span>";
    $breadcrumb[] = $elencoparole;
    $breadcrumb[] = "Search";
    if (!empty($breadcrumb))
      $breadcrumb = array_reverse($breadcrumb);
    drupal_set_breadcrumb($breadcrumb);
    $this->solrFilters = $islandora_fq;
    return;
  }
}
/**
 * Reset Results
 */
function resetResults()
{
  unset($this->solrResult);
}
/**
 * Execute the query
 * @return type
 */
function executeQuery()
{
  ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  // Right here the function should be split. One function to execute the query, //
  // another to set up pager variables and generate display. //
  ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  $url = variable_get('islandora_solr_search_block_url', 'http://localhost:8080/solr');
  $pathParts = parse_url($url);

```

```

    $solr = new Apache_Solr_Service($pathParts['host'], $pathParts['port'], $pathParts['path'] . '/');
    $solr->setCreateDocuments(0);
    $this->solrParams['hl'] = 'true';
    $this->solrParams['hl.fl'] = '*';
    $this->solrParams['hl.requireFieldMatch'] = 'true';
    $this->solrParams['hl.snippets'] = '10';
    $this->solrParams['hl.simple.pre'] = '<b>';
    $this->solrParams['hl.simple.post'] = '</b>';
    $bcksolrQuery = $this->solrQuery;
    $bcksolrQuery = preg_replace('~\s+~', ' ', trim($bcksolrQuery));
    $termini = explode(":", $bcksolrQuery);
    $parole = explode(" ", trim($termini[1]));
    $numeroparole = count($parole);
    $bcksolrQuery = "";
    $bcksolrQuery = "(" . $termini[0] . ":" . $parole[0] . ")";
    if ($numeroparole > 1)
    {
        for ($i = 1; $i < $numeroparole; $i++)
        {
            $bcksolrQuery = $bcksolrQuery . $termini[2] . "(" . $termini[0] . ":" . $parole[$i] . ")";
        }
    }
    // }
    // This is where the query gets executed and output starts being created.
    try
    {
        // $results = $solr->search($this->solrQuery, $this->solrStart, $this->solrLimit, $this->solrParams);
        $results = $solr->search($bcksolrQuery, $this->solrStart, $this->solrLimit, $this->solrParams);
    }
    catch (Exception $e)
    {
        drupal_set_message(t('error searching ') . $e->getMessage());
    }
    $this->solrResult = $results;
    ...
    
```



Picture 10: Multiple words full-text search example

2.9 Collection object for custom query and view

Using Islandora features we can produce custom views based on custom query. The steps are:

- Create Islandora standard collectionCModel object (islandora:collectionCModel).
- Add custom query into datastream QUERY
- Add custom view into datastream COLLECTION_VIEW
- Link repository object to drupal menu item

For example, to show repository statistics we created an object with these datastreams:

• DSID QUERY

```
select $ordine $colle $titlecolle
count (
  select $object
  from <#ri>
  where walk($object <fedora-rels-ext:isMemberOfCollection> $colle
  and $libro <fedora-rels-ext:isMemberOfCollection> $object)
  and $pagine <fedora-rels-ext:isMemberOf> $libro)

count (
  select $libro
  from <#ri>
  where walk($object <fedora-rels-ext:isMemberOfCollection> $colle
  and $libro <fedora-rels-ext:isMemberOfCollection> $object)
  and $libro <fedora-model:hasModel> <info:fedora/islandora:bookCModel>)

count (
  select $pagine
  from <#ri>
  where walk($object <fedora-rels-ext:isMemberOfCollection> $colle
  and $libro <fedora-rels-ext:isMemberOfCollection> $object)
  and $pagine <fedora-rels-ext:isMemberOf> $libro)

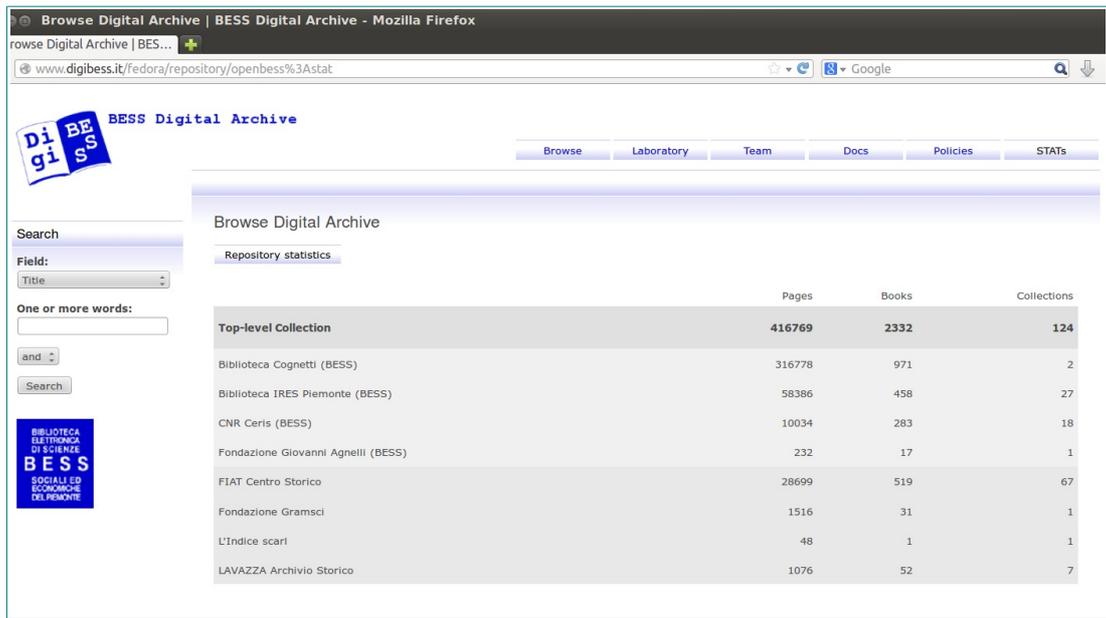
from <#ri>
where ((($colle <fedora-rels-ext:isMemberOfCollection> <info:fedora/islandora:root>
minus
$colle <mulgara:is> <info:fedora/openbess:collezioni>)
and $ordine <mulgara:is> '3')
or ($colle <fedora-rels-ext:isMemberOfCollection> <info:fedora/openbess:collezioni>
and $ordine <mulgara:is> '2')
or ($colle <mulgara:is> <info:fedora/islandora:root>
and $ordine <mulgara:is> '1'))
and $colle <dc:title> $titlecolle
order by $ordine $titlecolle
```

• DSID QUERYCOLLECTION_VIEW

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:s="http://www.w3.org/2001/sw/DataAccess/rfl/result"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:template match="/">
    <xsl:variable name="BASEURL">
      <xsl:value-of select="$baseUrl"/>
    </xsl:variable>
    <table>
      <tr>
        <td>
          <td>
          </td>
        <td align="right">
          Pages
        </td>
        <td align="right">
          Books
        </td>
        <td align="right">
          Collections
        </td>
      </tr>
      <xsl:for-each select="/s:sparql/s:results/s:result">
        <xsl:choose>
          <xsl:when test="s:ordine=1">
            <tr BGCOLOR="#E0E0E0" HEIGHT="50">
              <td>
                <b>
                  <xsl:value-of select="s:titlecolle"/>
                </b>
              </td>
              <td align="right">
                <b>
                  <xsl:value-of select="s:k2"/>
                </b>
              </td>
              <td align="right">
                <b>
                  <xsl:value-of select="s:k1"/>
                </b>
              </td>
              <td align="right">
                <b>
                  <xsl:value-of select="s:k0"/>
                </b>
              </td>
            </tr>
          </xsl:when>
        </xsl:choose>
      </xsl:for-each>
    </table>
```

```

</tr>
</xsl:when>
<xsl:when test="s:ordine=2">
<tr BGCOLOR="#F0F0F0" HEIGHT="35">
<td><xsl:value-of select="s:titlecolle"/> (BESS)
</td>
<td align="right">
<xsl:value-of select="s:k2"/>
</td>
<td align="right">
<xsl:value-of select="s:k1"/>
</td>
<td align="right">
<xsl:value-of select="s:k0"/>
</td>
</tr>
</xsl:when>
<xsl:when test="s:ordine=3">
<tr BGCOLOR="#E8E8E8" HEIGHT="35">
<td>
<xsl:value-of select="s:titlecolle"/>
</td>
<td align="right">
<xsl:value-of select="s:k2"/>
</td>
<td align="right">
<xsl:value-of select="s:k1"/>
</td>
<td align="right">
<xsl:value-of select="s:k0"/>
</td>
</tr>
</xsl:when>
</xsl:choose>
</xsl:for-each>
</table>
</xsl:template>
</xsl:stylesheet>
    
```



The screenshot shows the BESS Digital Archive website interface. On the left, there is a search bar with the field set to 'Title' and a search button. The main content area displays 'Browse Digital Archive' and 'Repository statistics'. A table lists various collections with their respective page counts, book counts, and collection counts.

	Pages	Books	Collections
Top-level Collection	416769	2332	124
Biblioteca Cognetti (BESS)	316778	971	2
Biblioteca IRES Piemonte (BESS)	58386	458	27
CNR Ceris (BESS)	10034	283	18
Fondazione Giovanni Agnelli (BESS)	232	17	1
FIAT Centro Storico	28699	519	67
Fondazione Gramsci	1516	31	1
L'Indice scari	48	1	1
LAVAZZA Archivio Storico	1076	52	7

Picture 11: Custom object for statistics view example

3 Repository

Regarding Back-end server which hosts repository and some more tomcat services, we used Fedora Commons feature to generate on-the-fly virtual datastream from datastream stored into the repository object.

3.1 Add PICO metadata to OAI-PMH

To achieve the purpose we have to ingest dissemination objects into the repository, add dissemination method to Book model and edit Oaiprovider (Proai) configuration. The method is based on Saxon web service.

- Ingest new object Sdef (PID *openbess:dc2picoSdef*) with these DS:

DC

```
<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/" xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
<dc:title>openbess:dc2picoSdef</dc:title>
<dc:identifier>openbess:dc2picoSdef</dc:identifier>
</oai_dc:dc>
```

RELS-EXT

```
<rdf:RDF xmlns:fedora-model="info:fedora/fedora-system:def/model#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#">
<rdf:Description rdf:about="info:fedora/openbess:dc2picoSdef">
<fedora-model:hasModel rdf:resource="info:fedora/fedora-system:ServiceDefinition-3.0"/>
</rdf:Description>
</rdf:RDF>
```

METHODMAP

```
<fmm:MethodMap xmlns:fmm="http://fedora.com/nsdlib.org/service/methodmap" name="methodmap">
<fmm:Method operationName="dc2pico"/>
</fmm:MethodMap>
```

- Ingest new object Sdep (PID *openbess:dc2picoSdep-bookCModel*) with these DS:

DC

```
<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/" xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
<dc:title>openbess:dc2picoSdep-bookCModel</dc:title>
<dc:identifier>openbess:dc2picoSdep-bookCModel</dc:identifier>
</oai_dc:dc>
```

RELS-EXT

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<rdf:Description rdf:about="info:fedora/openbess:dc2picoSdep-bookCModel">
<hasModel xmlns="info:fedora/fedora-system:def/model#" rdf:resource="info:fedora/fedora-
system:ServiceDeployment-3.0"></hasModel>
<isDeploymentOf xmlns="info:fedora/fedora-system:def/model#"
rdf:resource="info:fedora/openbess:dc2picoSdef"></isDeploymentOf>
<isContractorOf xmlns="info:fedora/fedora-system:def/model#"
rdf:resource="info:fedora/islandora:bookCModel"></isContractorOf>
</rdf:Description>
</rdf:RDF>
```

DSINPUTSPEC

```
<fbs:DSInputSpec xmlns:fbs="http://fedora.com/nsdlib.org/service/bindspec" label="dc2picoSdepInputSpec">
<fbs:DSInput DSMax="1" DSMin="1" DSOrdinality="false" wsdlMsgPartName="DC">
<fbs:DSInputLabel>DC</fbs:DSInputLabel>
<fbs:DSMIME>text/xml</fbs:DSMIME>
<fbs:DSInputInstruction/>
</fbs:DSInput>
<fbs:DSInput DSMax="1" DSMin="1" DSOrdinality="false" pid="openbess:dc2picoSdep-bookCModel"
wsdlMsgPartName="XSL">
<fbs:DSInputLabel>XSL</fbs:DSInputLabel>
<fbs:DSMIME>text/xml</fbs:DSMIME>
<fbs:DSInputInstruction/>
</fbs:DSInput>
</fbs:DSInputSpec>
```

WDSL

```
<wsdl:definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap"
xmlns:soapenc="http://schemas.xmlsoap.org/wsdl/soap/encoding" xmlns:this="dc2picoSdep"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="dc2picoSdep"
targetNamespace="dc2picoSdep">
```

```

<wsdl:message name="dc2picoRequestMsg">
  <wsdl:part name="DC" type="xsd:string"/>
  <wsdl:part name="XSL" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="response">
  <wsdl:part name="exhibit" type="xsd:base64Binary"/>
</wsdl:message>
<wsdl:portType name="dc2pico_portType">
  <wsdl:operation name="dc2pico">
    <wsdl:input message="this:dc2picoRequestMsg"/>
    <wsdl:output message="this:response"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:service name="dc2pico_service">
  <wsdl:port binding="this:dc2pico_binding" name="dc2pico_port">
    <http:address location="http://fc1.to.cnr.it:8080/saxon"/>
  </wsdl:port>
</wsdl:service>
<wsdl:binding name="dc2pico_binding" type="this:dc2pico_portType">
  <http:binding verb="GET"/>
  <wsdl:operation name="dc2pico">
    <http:operation location="SaxonServlet?source=(DC) &amp;style=(XSL) &amp;clear-stylesheet-cache=yes"/>
    <wsdl:input>
      <http:urlReplacement/>
    </wsdl:input>
    <wsdl:output>
      <mime:content type="text/xml"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
</wsdl:definitions>
    
```

METHODMAP

```

<fmm:MethodMap xmlns:fmm="http://fedora.com/nsdlib.org/service/methodmap" name="methodmap">
  <fmm:Method operationName="dc2pico" wsdlMsgName="dc2picoRequestMsg" wsdlMsgOutput="response">
    <fmm:DataStreamInputParm paramName="DC" passBy="URL_REF" required="true"/>
    <fmm:DataStreamInputParm paramName="XSL" passBy="URL_REF" required="true"/>
    <fmm:UserInputParm defaultValue="" paramName="uid" passBy="VALUE" required="false"/>
    <fmm:MethodReturnType wsdlMsgName="response" wsdlMsgTOMIME="text/xml"/>
  </fmm:Method>
</fmm:MethodMap>
    
```

XSL (mapping between Dublin Core and PICO metadata)

```

<xsl:stylesheet xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:fn="http://www.w3.org/2005/xpath-functions" xmlns:foxml="info:fedora/fedora-system:def/foxml#"
  xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/" xmlns:pico="http://purl.org/pico/1.0/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform" exclude-result-prefixes="oai_dc fn foxml rdf" version="1.0"
  xsi:schemaLocation="http://purl.org/pico/1.0/
  http://www.culturaitalia.it/opencms/export/sites/culturaitalia/attachments/schemas/1.0/pico.xsd">
  <xsl:output encoding="UTF-8" indent="yes" method="xml"/>
  <xsl:variable name="ns" select="'http://purl.org/pico/1.0/'"/>
  <xsl:variable name="schemaLoc" select="'http://purl.org/pico/1.0/
  http://www.culturaitalia.it/opencms/export/sites/culturaitalia/attachments/schemas/1.0/pico.xsd'"/>
  <xsl:template match="/">
    <pico:record>
      <xsl:attribute name="xsi:schemaLocation" namespace="http://www.w3.org/2001/XMLSchema-instance">
        <xsl:value-of select="$schemaLoc"/>
      </xsl:attribute>
      <dc:title>
        <xsl:value-of select="/oai_dc:dc/dc:title"/>
      </dc:title>
      <dc:creator>
        <xsl:value-of select="/oai_dc:dc/dc:creator"/>
      </dc:creator>
      <xsl:apply-templates select="/oai_dc:dc/dc:subject"/>
      <xsl:if test="/oai_dc:dc/dc:description != ''">
        <dc:description>
          <xsl:value-of select="/oai_dc:dc/dc:description"/>
        </dc:description>
      </xsl:if>
      <dc:publisher>
        <xsl:value-of select="/oai_dc:dc/dc:publisher"/>
      </dc:publisher>
      <dc:date>
        <xsl:value-of select="/oai_dc:dc/dc:date"/>
      </dc:date>
      <xsl:apply-templates select="/oai_dc:dc/dc:type"/>
      <xsl:for-each select="/oai_dc:dc/dc:identifier">
        <xsl:choose>
          <xsl:when test="contains(., 'http')">
            <xsl:element name="dcterms:isReferencedBy">
              <xsl:attribute name="xsi:type" namespace="http://www.w3.org/2001/XMLSchema-
  instance">pico:Anchor</xsl:attribute>
              <xsl:value-of select="'title=consulta la scheda esterna;URL='"/>
              <xsl:value-of select="."/>
            </xsl:element>
            <xsl:element name="pico:preview">
              <xsl:attribute name="xsi:type" namespace="http://www.w3.org/2001/XMLSchema-
  instance">dcterms:URI</xsl:attribute>
              <xsl:value-of select="."/>
              <xsl:value-of select="'/TN'"/>
            </xsl:element>
          </xsl:when>
        </xsl:choose>
      </xsl:for-each>
    </pico:record>
  </xsl:template>
    
```

```

        <xsl:otherwise>
          <dc:identifier>
            <xsl:value-of select="."/>
          </dc:identifier>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:for-each>
  </dc:language>
  <xsl:value-of select="/oai_dc:dc:language"/>
</dc:language>
</pico:record>
</xsl:template>
<xsl:template match="dc:subject">
  <dc:subject>
    <xsl:value-of select="/oai_dc:dc:subject"/>
  </dc:subject>
  <xsl:element name="dc:subject">
    <xsl:attribute name="xsi:type" namespace="http://www.w3.org/2001/XMLSchema-instance">pico:Thesaurus</xsl:attribute>
    <xsl:value-of select="'http://culturaitalia.it/pico/thesaurus/4.2#libri'"/>
  </xsl:element>
</xsl:template>
<xsl:template match="dc:type">
  <xsl:element name="dc:type">
    <xsl:attribute name="xsi:type" namespace="http://www.w3.org/2001/XMLSchema-instance">dcterms:DCMIType</xsl:attribute>
    <xsl:value-of select="/oai_dc:dc:type"/>
  </xsl:element>
</xsl:template>
</xsl:stylesheet>

```

- Edit Book model (PID *islandora:bookCModel*) and add the service to RELS-EXT datastream

```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  <rdf:Description rdf:about="info:fedora/islandora:bookCModel">
    <hasModel xmlns="info:fedora/fedora-system:def/model#" rdf:resource="info:fedora/fedora-system:ContentModel-3.0"/>
    <hasService xmlns="info:fedora/fedora-system:def/model#" rdf:resource="info:fedora/islandora:viewerSdef"/>
    <hasService xmlns="info:fedora/fedora-system:def/model#" rdf:resource="info:fedora/openbess:dc2picoSdef"/>
  </rdf:Description>
</rdf:RDF>

```

- Edit oaiprovider configuration and add new PICO metadata format

```

...
driver.fedora.md.formats = oai_dc pico
driver.fedora.md.format.oai_dc.loc = http://www.openarchives.org/OAI/2.0/oai_dc.xsd
driver.fedora.md.format.pico.loc =
  http://www.culturaitalia.it/opencms/export/sites/culturaitalia/attachments/schemas/1.0/pico.xsd
driver.fedora.md.format.oai_dc.uri = http://www.openarchives.org/OAI/2.0/oai_dc/
driver.fedora.md.format.pico.uri = http://purl.org/pico/1.0/
driver.fedora.md.format.oai_dc.dissType = info:fedora/*/DC
driver.fedora.md.format.pico.dissType = info:fedora/*/openbess:dc2picoSdef/dc2pico
...

```

- OAI-PMH harvesting example

URL

```

http://fcl.to.cnr.it/oaiprovider/
?verb=GetRecord&identifier=oai:fcl.to.cnr.it:openbess:T0043-00057&metadataPrefix=pico

```

return

```

<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
  http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2013-09-25T08:46:09Z</responseDate>
  <request verb="GetRecord" identifier="oai:fcl.to.cnr.it:openbess:T0043-00057"
  metadataPrefix="pico">http://fcl.to.cnr.it/oaiprovider/</request>
  <GetRecord>
    <record>
      <header>
        <identifier>oai:fcl.to.cnr.it:openbess:T0043-00057</identifier>
        <timestamp>2013-03-15T12:14:57Z</timestamp>
        <setSpec>digibess:openbess:cognetti:directors</setSpec>
      </header>
      <metadata>
        <pico:record xmlns:pico="http://purl.org/pico/1.0/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:dcterms="http://purl.org/dc/terms/"
  xsi:schemaLocation="http://purl.org/pico/1.0/
  http://www.culturaitalia.it/opencms/export/sites/culturaitalia/attachments/schemas/1.0/pico.xsd">
          <dc:title>Monografie di soci e allievi del laboratorio di economia politica : 1909-911</dc:title>
          <dc:creator>R. Università di Torino</dc:creator>
          <dc:subject>Economia, lavoro, società</dc:subject>
          <dc:subject xsi:type="pico:Thesaurus">http://culturaitalia.it/pico/thesaurus/4.2#libri</dc:subject>
          <dc:publisher>s.n.</dc:publisher>
          <dc:date>1909</dc:date>
          <dc:type xsi:type="dcterms:DCMIType">Text</dc:type>
          <dc:identifier>openbess:T0043-00057</dc:identifier>

```

```

<dcterms:isReferencedBy xsi:type="pico:Anchor">title=consulta la scheda
esterna;URL=http://www.digibess.it/fedora/repository/openbess:T0043-00057</dcterms:isReferencedBy>
<pico:preview xsi:type="dcterms:URI">http://www.digibess.it/fedora/repository/openbess:T0043-
00057/TN</pico:preview>
<dc:language>it</dc:language>
</pico:record>
</metadata>
</record>
</GetRecord>
</OAI-PMH>
    
```

3.2 Dissemination method for Low-Res JPEG from TIFF

In this case we have to ingest dissemination objects into the repository and add dissemination method to Page model. The method is based on Adore-djatoka web service.

- Ingest new object Sdef (PID *openbess:jpgldSdef*) with these DS:

DC

```

<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/" xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
<dc:title>openbess:jpgldSdef</dc:title>
<dc:identifier>openbess:jpgldSdef</dc:identifier>
</oai_dc:dc>
    
```

RELS-EXT

```

<rdf:RDF xmlns:fedora-model="info:fedora/fedora-system:def/model#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#">
<rdf:Description rdf:about="info:fedora/openbess:jpgldSdef">
<fedora-model:hasModel rdf:resource="info:fedora/fedora-system:ServiceDefinition-3.0"/>
</rdf:Description>
</rdf:RDF>
    
```

METHODMAP

```

<fmm:MethodMap xmlns:fmm="http://fedora.com/nsdlib.org/service/methodmap" name="methodmap">
<fmm:Method operationName="getJpgLD"/>
</fmm:MethodMap>
    
```

- Ingest new object Sdep (PID *openbess:jpgldSdep-pageCModel*) with these DS:

DC

```

<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/" xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
<dc:title>openbess:jpgldSdep-pageCModel</dc:title>
<dc:identifier>openbess:jpgldSdep-pageCModel</dc:identifier>
</oai_dc:dc>
    
```

RELS-EXT

```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<rdf:Description rdf:about="info:fedora/openbess:jpgldSdep-pageCModel">
<hasModel xmlns="info:fedora/fedora-system:def/model#" rdf:resource="info:fedora/fedora-
system:ServiceDeployment-3.0"/>
<isDeploymentOf xmlns="info:fedora/fedora-system:def/model#" rdf:resource="info:fedora/openbess:jpgldSdef"/>
<isContractorOf xmlns="info:fedora/fedora-system:def/model#"
rdf:resource="info:fedora/islandora:pageCModel"/>
</rdf:Description>
</rdf:RDF>
    
```

DSINPUTSPEC

```

<fbs:DSInputSpec xmlns:fbs="http://fedora.com/nsdlib.org/service/bindspec" label="jp2SdepInputSpec">
<fbs:DSInput DSMax="1" DSMin="1" DSOrdinality="false" wsdlMsgPartName="DC">
<fbs:DSInputLabel>DC</fbs:DSInputLabel>
<fbs:DSMIME>text/xml</fbs:DSMIME>
<fbs:DSInputInstruction/>
</fbs:DSInput>
</fbs:DSInputSpec>
    
```

WDSL

```

<wsdl:definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap"
xmlns:soapenc="http://schemas.xmlsoap.org/wsdl/soap/encoding" xmlns:this="djatoka"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="Fedora Local
Djatoka Service" targetNamespace="jpgldSdep">
<wsdl:message name="getRegionRequest">
<wsdl:part name="PID" type="xsd:string"/>
<wsdl:part name="DSID" type="xsd:string"/>
<wsdl:part name="svc_id" type="xsd:string"/>
    
```

```

</wsdl:message>
<wsdl:message name="response">
  <wsdl:part name="exhibit" type="xsd:base64Binary"/>
</wsdl:message>
<wsdl:portType name="FedoraDjatokaPortType">
  <wsdl:operation name="getJpgLD">
    <wsdl:input message="this:getRegionRequest"/>
    <wsdl:output message="this:response"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:service name="FedoraDjatoka">
  <wsdl:port binding="this:FedoraDjatoka_http" name="FedoraDjatoka_port">
    <http:address location="http://fc1.to.cnr.it:8080"/>
  </wsdl:port>
</wsdl:service>
<wsdl:binding name="FedoraDjatoka_http" type="this:FedoraDjatokaPortType">
  <http:binding verb="GET"/>
  <wsdl:operation name="getJpgLD">
    <http:operation location="adore-djatoka/resolver?url_ver=Z39.88-
2004&amp;rft_id=http://fc1.to.cnr.it:8080/fedora/get/(PID)/
(DSID)&amp;svc_id=(svc_id)&amp;svc_val_fmt=info:ofi/fmt:kev:mtx:jpeg2000&amp;svc.format=image/jpeg&amp;svc.lev
el=4"/>
    <wsdl:input>
      <http:urlReplacement/>
    </wsdl:input>
    <wsdl:output>
      <mime:content type="image/jpeg"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
</wsdl:definitions>

```

METHODMAP

```

<fmm:MethodMap xmlns:fmm="http://fedora.com/nsdlib.org/service/methodmap" bDefPID="djatoka:bDef" name="MethodMap
- djatoka Service Methods">
  <fmm:Method operationName="getJpgLD" wsdlMsgName="getRegionRequest" wsdlMsgOutput="response">
    <fmm:DefaultInputParm defaultValue="$PID" label="fedora object pid" paramName="PID" passBy="VALUE"
required="true"/>
    <fmm:DefaultInputParm defaultValue="TIFF" label="content model" paramName="DSID" passBy="VALUE"
required="true"/>
    <fmm:DefaultInputParm defaultValue="info:lanl-repo/svc/getRegion" paramName="svc_id" passBy="VALUE"
required="true"/>
    <fmm:MethodReturnType wsdlMsgName="response" wsdlMsgTOIMIME="image/jpeg"/>
  </fmm:Method>
</fmm:MethodMap>

```

- Edit Page model (PID islandora:pageCModel) and add the service to RELS-EXT datastream

```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description rdf:about="info:fedora/islandora:pageCModel">
    <hasModel xmlns="info:fedora/fedora-system:def/model#" rdf:resource="info:fedora/fedora-system:ContentModel-
3.0"/>
    <hasService xmlns="info:fedora/fedora-system:def/model#" rdf:resource="info:fedora/islandora:viewerSdef"/>
    <hasService xmlns="info:fedora/fedora-system:def/model#" rdf:resource="info:fedora/islandora:jp2Sdef"/>
    <hasService xmlns="info:fedora/fedora-system:def/model#" rdf:resource="info:fedora/islandora:tei2htmlSdef"/>
    <hasService xmlns="info:fedora/fedora-system:def/model#" rdf:resource="info:fedora/openbess:jpgldSdef"/>
  </rdf:Description>
</rdf:RDF>

```

4 Managing and ingesting scripts

Here some useful scripts we use to manage existing objects, prepare and ingest documents into the repository.

4.1 Script to modify DC datastream dc:date

This procedure is based on *fedora-modify.sh* included in Fedora Commons client side scripts which allows to modify an existing datastream but not a single element of DC metadata.

The main script *modificaDSbookepage.sh* calls php script *modificaDCdate.php* which reads current DC datastream and prepare directive file for *fedora-modify.sh*. The main script invokes Fedora Commons client script to modify DC metadata and it repeats these steps for each page of the book object.

- *modificaDSbookepage.sh* (main script)

```
#!/bin/bash
#
# Input parameters
# 1 Book PID
# 2 New date
# 3 N pages
# 4 fedoraAdmin password
#
#example ./modificaDSbookepage.sh fgramscl:TO024-00001 1984-01 52 password
#
PID=$1
NEWDATE=$2
PAGES=$3
./modificaDCdate.php $PID $NEWDATE
risultato=$(./fedora-modify.sh fcl.to.cnr.it:8080 fedoraAdmin $4 bmodify.xml ../logs/log http)
if [[ "$risultato" == *Valid* ]]
then
    echo "Is Valid ..."
    ./fedora-modify.sh fcl.to.cnr.it:8080 fedoraAdmin $4 bmodify.xml ../logs/log http YES
else
    echo "ERROR XML: $risultato"
    exit
fi
for (( npage=1; npage<=$PAGES; npage++ ))
do
    snpage=$(printf "%04d" $npage)
    pagePID="$PID"-"$snpage"
    echo $pagePID
    ./modificaDCdate.php $pagePID $NEWDATE
    risultato=$(./fedora-modify.sh fcl.to.cnr.it:8080 fedoraAdmin $4 bmodify.xml ../logs/log http)
    if [[ "$risultato" == *Valid* ]]
    then
        echo "Is Valid ..."
        ./fedora-modify.sh fcl.to.cnr.it:8080 fedoraAdmin $4 bmodify.xml ../logs/log http YES
    else
        echo "ERROR XML: $risultato"
        exit
    fi
done
exit
```

- *modificaDCdate.php*

```
#!/usr/bin/php
<?php
if ($argc != 3 || in_array($argv[1], array('--help', '-help', '-h', '-?'))) {
?>
    This is a command line PHP script with two options.
    Usage:
<?php
    echo $argv[0]; ?> <option>

    <option> can be some word you would like
    to print out. With the --help, -help, -h,
    or -? options, you can get this help.

<?php
    exit;
} else {
    $PID = $argv[1];
    $Nuovadata = $argv[2];
}
$indoc = new DOMDocument();
$indoc->formatOutput = true;
$indoc->preserveWhiteSpace = false;
$indoc->load('http://fcl.to.cnr.it:8080/fedora/objects/' . $PID . '/datastreams/DC/content');
$innode = $indoc->getElementsByTagNameNS('http://www.openarchives.org/OAI/2.0/oai_dc/', 'dc')->item(0);
```

```

$dcdate = $innode->getElementsByTagNameNS('http://purl.org/dc/elements/1.1/', 'date')->item(0);
$doldddate = $innode->removeChild($dcdate);
print_r($dcdate->nodeValue . " <- " . $Nuovadata . "\n");
$newdate = $indoc->createElement('dc:date', $Nuovadata);
$innode->appendChild($newdate);
$dom = new DOMDocument('1.0', 'utf-8');
$dom->formatOutput = true;
$dom->preserveWhiteSpace = false;
$element = $dom->createElement('fbm:batchModify');
$element->setAttribute('xmlns:fbm', 'http://www.fedora.info/definitions/');
$element->setAttribute('xmlns:xsi', 'http://www.w3.org/2001/XMLSchema-instance');
$element->setAttribute('xsi:schemaLocation', 'http://www.fedora.info/definitions/ http://www.fedora-
commons.org/definitions/1/0/api/batchModify-1.1.xsd');
$dom->appendChild($element);
$element1 = $dom->createElement('fbm:modifyDatastream');
$element1->setAttribute('pid', $PID);
$element1->setAttribute('dsID', 'DC');
$element1->setAttribute('dsControlGroupType', 'X');
$element1->setAttribute('logMessage', 'BatchModify - modifyDatastream');
$element->appendChild($element1);
$element2 = $dom->createElement('fbm:xmlData');
$element1->appendChild($element2);
$newnode = $dom->importNode($innode, true);
$element2->appendChild($newnode);
print 'Wrote: ' . $dom->save("bmodify.xml") . ' bytes' . "\n";
?>

```

- **bmodify.xml example**

```

<?xml version="1.0" encoding="utf-8"?>
<fbm:batchModify xmlns:fbm="http://www.fedora.info/definitions/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.fedora.info/definitions/ http://www.fedora-
commons.org/definitions/1/0/api/batchModify-1.1.xsd">
  <fbm:modifyDatastream pid="fgramsci:T0024-00008-0056" dsID="DC" dsControlGroupType="X" logMessage="BatchModify
- modifyDatastream">
    <fbm:xmlData>
      <oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
        <dc:title>Sisifo 8 - page 0056</dc:title>
        <dc:creator>Istituto Gramsci piemontese</dc:creator>
        <dc:subject>sviluppo; industria; welfare; lavoro; sociale; ricerca; innovazione; cultura, Piemonte,
Torino</dc:subject>
        <dc:description>idee, ricerche, programmi dell'Istituto Gramsci piemontese. Settembre
1986</dc:description>
        <dc:publisher>Istituto Gramsci piemontese</dc:publisher>
        <dc:type>Text</dc:type>
        <dc:format/>
        <dc:identifier>fgramsci:T0024-00008-0056</dc:identifier>
        <dc:language>it</dc:language>
        <dc:date>1986-09</dc:date>
      </oai_dc:dc>
    </fbm:xmlData>
  </fbm:modifyDatastream>
</fbm:batchModify>

```

4.2 Generate BOOK files from PDF

This procedure describe how to generate all files needed for Book and Page ingesting, starting from a PDF file (pdf/a).

- OCR

The first step is the OCR extraction from PDF file. It is achieved on a desktop computer with LibreOffice and Dcspplit installed. Run custom *pdfTOtxt.sh* script with absolute path to pdf file directory as parameter.

```

#!/bin/bash

bookdir=$1

echo "$bookdir"
cd "$bookdir"
n=0
SAVEIFS=$IFS
IFS=$(echo -en "\n\b")
for nfile in $(find *.pdf -type f);
do
  let n += 1
  filedoc="$nfile"
done
if [ $n -gt 1 ] || [ $n -lt 1 ]
then
  echo "ERROR file PDF non unico"
  exit
fi

```

```

IFS=$SAVEIFS
cp "$filedoc" doc.pdf
rm "$filedoc"
docsplit text --pages all --no-ocr --no-clean --output OCR/ doc.pdf
cd OCR
n=0
SAVEIFS=$IFS
IFS=$(echo -en "\n\b")
for nfile in $(find *.txt -type f);
do
    numer=${nfile#doc_}
    numero=${numer%\.Ext}
    sn=$(printf "%04d" $numero)

    tr -d '\f' < "$nfile" > "$sn".txt
    rm "$nfile"

    echo "$sn" "DONE"
done
IFS=$SAVEIFS
cd ~
echo "DONE *****"$bookdir"
exit
    
```

The script creates OCR directory with a single txt file for each pdf page (i.e. 0001.txt, 0002.txt, ...). Book directory containing pdf and txt files can now be moved to back-end server.

- TIFF

The next step is achieved on back-end server and produces tiff image for each single pdf page using ImageMagick and pdftk tools. Run custom *pdfatiff.sh* script with absolute path to book directory as parameter.

```

#!/bin/bash
bookdir=$1

echo "$bookdir"
n=0
SAVEIFS=$IFS
IFS=$(echo -en "\n\b")
for nfile in $(find "$bookdir"/*.pdf -type f);
do
    let "n += 1"
    filepdf="$nfile"
done
if [ $n -gt 1 ] || [ $n -lt 1 ]
then
    echo "ERROR file PDF non unico"
    exit
fi
mkdir "$bookdir"/pdfs
cp "$filepdf" "$bookdir"/pdfs
cd "$bookdir"/pdfs
pdftk "$filepdf" burst output pg-%04d.pdf
n=0
SAVEIFS=$IFS
IFS=$(echo -en "\n\b")
for nfile in $(find pg-*.pdf -type f);
do
    let "n += 1"
    sn=$(printf "%04d" $n)
    filepdf="$nfile"
    echo "$filepdf" -> "$sn.tif"
    pdftk "$filepdf" output "temp.pdf"
    # For PDF from image
    # convert -density 150 "temp.pdf" "$sn.tif"
    # For PDF from Word
    convert -background white -flatten -density 600 -resize 1200 -border 0.5% -bordercolor LightGray "temp.pdf"
    ../"$sn.tif"
    rm "temp.pdf"
done
cd ~/clineFC
rm -R "$bookdir"/pdfs
exit
    
```

Book is now ready for ingesting.

4.3 Book ingesting script update

The updated versions of Book and Page ingesting scripts are available in the Appendix.
For a complete description refer to Step by Step Guide part 1.

5 Webography

- [1] Apache Solr, <http://lucene.apache.org/solr/> , visited September 2013.
- [2] Apache Tomcat, <http://tomcat.apache.org/> , visited September 2013.
- [3] CNR Ismar, BSA digital Library, <http://biblio1.ve.ismar.cnr.it/> , visited September 2013.
- [4] DigiBESS repository, www.digibess.it visited September 2013.
- [5] Docsplit <http://documentcloud.github.com/docsplit/> visited September 2013.
- [6] Drupal Come for the software, stay for the community, <http://drupal.org/> , visited September 2013.
- [7] Dublin Core Metadata Initiative, <http://dublincore.org/> , visited September 2013.
- [8] Fedora Commons software repository, <http://fedora-commons.org/> , visited September 2013
- [9] Infrastruttura di rete del CNR in Piemonte, <http://www.to.cnr.it> , visited September 2013.
- [10] Islandora, building a rich digital repository ecosystem, <http://islandora.ca/> , visited September 2013.
- [11] Islandora developers forum, <https://github.com/islandora> , visited September 2013.
- [12] OAI-PMH validator, <http://validator.oaipmh.com/> , visited September 2013.
- [13] OAI-PMH The Open Archives Initiative Protocol for Metadata Harvesting, <http://www.openarchives.org/OAI/openarchivesprotocol.html> , visited September 2013.
- [14] Rapporto tecnico Ceris-CNR n.37 Storage in HA: cluster attivo/passivo open source. Giancarlo Birello, Ivano Fucile, Valter Giovanetti, Anna Perin, June 2011, <http://www.digibess.it/fedora/repository/openbess:TO094-00158> , visited September 2013.
- [15] Rapporto tecnico Ceris-CNR n. 39 Repository e Front-end open-source per la conservazione di opere digitali . Giancarlo Birello, Ivano Fucile, Valter Giovanetti, Anna Perin, November 2011 <http://www.digibess.it/fedora/repository/openbess:TO094-00160> , visited September 2013.
- [16] Rapporto tecnico Ceris-CNR n.41 Storage in HA: manutenzione ordinaria e straordinaria . Giancarlo Birello, Ivano Fucile, Valter Giovanetti, Anna Perin, February 2012. <http://www.digibess.it/fedora/repository/openbess:TO094-00162> , visited September 2013.
- [17] Rapporto tecnico Ceris-CNR n.42 "Step by step Installation Guide of a Digital Preservation Infrastructure.", Giancarlo Birello, Ivano Fucile, Valter Giovanetti, Anna Perin, maggio 2012. <http://www.digibess.it/fedora/repository/openbess:TO094-00163> , visited September 2013.
- [18] Rapporto tecnico Ceris-CNR n.44 "KVM: an open-source framework for virtualization.", Giancarlo Birello, Ivano Fucile, Valter Giovanetti, Anna Perin, September 2013. <http://www.digibess.it/fedora/repository/openbess:TO094-00284> , visited September 2013.

6 Appendix

6.1 Front-end Server

6.1.1 sites/all/modules/islandora-github/ObjectHelper.inc

```

<?php
// $Id$
/**
 * @file
 * Object Helper Class
 */

/**
 * ObjectHelper Class ??
 */
class ObjectHelper {

//allowed operations
public static $OBJECT_HELPER_VIEW_FEDORA = 'view fedora collection';
public static $EDIT_FEDORA_METADATA = 'edit fedora meta data';
public static $PURGE_FEDORA_OBJECTSANDSTREAMS = 'purge objects and datastreams';
public static $ADD_FEDORA_STREAMS = 'add fedora datastreams';
public static $INGEST_FEDORA_OBJECTS = 'ingest new fedora objects';
public static $EDIT_TAGS_DATASTREAM = 'edit tags datastream';
public static $VIEW_DETAILED_CONTENT_LIST = 'view detailed list of content';
public static $DISPLAY_ALWAYS = 0;
public static $DISPLAY_NEVER = 1;
public static $DISPLAY_NO_MODEL_OUTPUT = 2;
// TODO: Make this into a static member constant
public $availableDataStreamsText = 'Detailed list of content';

/**
 * Constructor
 */
function ObjectHelper() {
  drupal_bootstrap(DRUPAL_BOOTSTRAP_FULL);
  module_load_include('inc', 'fedora_repository', 'ConnectionHelper');
  $connectionHelper = new ConnectionHelper();
  //$this->fedoraUser = $connectionHelper->getUser();
  //$this->fedoraPass = $connectionHelper->getPassword();
}

/**
 * Grabs a stream from fedora sets the mimetype and returns it. $dsID is the
 * datastream id. If $forceSoap is set, the function will always buffer the datastream from fedora.
 * Otherwise, it will
 * try and use a redirect if possible.
 *
 * @global type $user
 * @param type $pid
 * @param type $dsID
 * @param type $asAttachment
 * @param type $label
 * @param type $filePath
 * @param type $version
 * @param type $forceSoap
 * @return type
 */
function makeObject($pid, $dsID, $asAttachment = FALSE, $label = NULL, $filePath=FALSE, $version=NULL,
$forceSoap = TRUE) {
  global $user;
  module_load_include('inc', 'fedora_repository', 'ContentModel');
  if ($pid == NULL || $dsID == NULL) {
    drupal_set_message(t("no pid or dsid given to create an object with"), 'error');
    return ' ';
  }
  $headers = module_invoke_all('file_download', "/fedora/repository/$pid/$dsID");
  if (in_array(-1, $headers)) {
    drupal_set_message(t('hello'));
    drupal_access_denied();
    return ' ';
  }
  if (!fedora_repository_access(OBJECTHELPER :: $OBJECT_HELPER_VIEW_FEDORA, $pid, $user)) {
    drupal_set_message(t("You do not have access Fedora objects within the attempted namespace."), 'error');
    drupal_access_denied();
    return ' ';
  }
  if (variable_get('fedora_object_restrict_datastreams', FALSE) == TRUE) {
    if (($cm = ContentModel::loadFromObject($pid)) == FALSE) {
      drupal_set_message(t("You do not have access to objects without an Islandora Content Model."), 'error');
      drupal_access_denied();
      return ' ';
    }
    $cmDatastreams = $cm->listDatastreams();
    if (!(isset($user) && in_array('administrator', $user->roles)) || in_array($dsID, $cmDatastreams)) {
      drupal_set_message(t("You do not have access to the specified datastream."), 'error');
      drupal_access_denied();
      return ' ';
    }
  }
}
}

```

```

module_load_include('inc', 'fedora_repository', 'api/fedora_item');
$item = new Fedora_Item($pid);
if (isset($item->datastreams[$dsID])) {
    $mimeType = $item->datastreams[$dsID]['MIMETYPE'];
    if ($label == NULL) {
        $label = $item->datastreams[$dsID]['label'];
    }
}
else if ($dsID == "openbess:jpgldSdef") {
    $mimeType = "image/jpeg";
}
else {
    drupal_not_found();
    exit();
}
if ((!isset($user)) || $user->uid == 0) {
    $fedoraUser = 'anonymous';
    $fedoraPass = 'anonymous';
    $contentType = 0;
}
else {
    $fedoraUser = $user->name;
    $fedoraPass = $user->pass;
    $dataStreamInfo = $item->get_datastream_info($dsID);
    $contentType = $dataStreamInfo->datastream->size;
}
if (function_exists("curl_init")) {
    if (!isset($mimeType)) {
        $pid = variable_get('fedora_default_display_pid', 'demo:10');
        $dsID = variable_get('fedora_default_display_dsID', 'TN');
        $mimeType = 'image/jpeg';
    }
    if ($dsID == "openbess:jpgldSdef") {
        $url = variable_get('fedora_base_url', 'http://localhost:8080/fedora') . '/get/' . $pid .
        '/openbess:jpgldSdef/getJpgLD';
    }
    else {
        $url = variable_get('fedora_base_url', 'http://localhost:8080/fedora') . '/get/' . $pid . '/' . $dsID;
    }
    if ($version) {
        $url .= '/' . $version; //drupal_urlencode($version);
    }
    $ch = curl_init();
    $user_agent = "Mozilla/4.0 pp(compatible; MSIE 5.01; Windows NT 5.0)";
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, FALSE);
    curl_setopt($ch, CURLOPT_FAILONERROR, 1); // Fail on errors
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1); // allow redirects
    //curl_setopt($ch, CURLOPT_TIMEOUT, 15); // times out after 15s
    curl_setopt($ch, CURLOPT_USERAGENT, $user_agent);
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_USERPWD, "$fedoraUser:$fedoraPass");
    // There seems to be a bug in Fedora 3.1's REST authentication, removing this line fixes the authorization
    denied error.
    // curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_ANY);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 0); // return into a variable
    curl_setopt($ch, CURLOPT_URL, $url);
    if ($filePath != FALSE) {
        $fp = fopen($filePath, 'w');
        curl_setopt($ch, CURLOPT_FILE, $fp);
        curl_setopt($ch, CURLOPT_HEADER, 0);
        curl_exec($ch);
        fclose($fp);
    }
    else {
        header("Content-type: $mimeType");
        if (($contentType > 0) && ($dsID != "openbess:jpgldSdef")) {
            header("Content-length: $contentType");
        }
        if ($asAttachment) {
            $suggestedFileName = "$label";
            $pos = strpos($suggestedFileName, '.');
            /*
            * Here we used to take an object of, say, type application/pdf with label, say, "My Document"
            * and we assemble the output filename extension based on the post-slash portion of the mimetype.
            * (If the label has a period anywhere in it, we leave it alone.)
            *
            * This is great for simple mimetypes like application/pdf, text/html, image/jpeg, etc.
            * but it's terrible for, say, application/vnd.oasis.opendocument.text (.odt).
            *
            * Instead we'll use the get_extension function in MimeClass.inc to discover a valid extension
            * for the mimetype in question.
            */
            if ($pos === FALSE) {
                module_load_include('inc', 'fedora_repository', 'MimeClass');
                $mimeClass = new MimeClass();
                $ext = $mimeClass->get_extension($mimeType);
                $suggestedFileName = "$label.$ext";
            }
            header('Content-Disposition: attachment; filename="' . $suggestedFileName . '"');
        }
        if ((isset($user) && $user->uid != 0) || $forceSoap || isset($_SERVER['HTTPS'])) {
            curl_exec($ch);
        }
        else {

```

```

        header('Location: ' . $url);
    }
    curl_close($ch);
}
else {
    drupal_set_message(t('No curl support.'), 'error');
}
}

/**
 * Gets collection objects t
 *
 * @param type $pid
 * @param type $query
 * @return type
 */
function getCollectionInfo($pid, $query = NULL) {
    module_load_include('inc', 'fedora_repository', 'CollectionClass');
    $collectionClass = new CollectionClass();
    $results = $collectionClass->getRelatedItems($pid, $query);
    return $results;
}

/**
 * returns the mime type
 *
 * @global type $user
 * @param type $pid
 * @param type $dsID
 * @return type
 */
function getMimeType($pid, $dsID) {
    global $user;
    if (empty($pid) || empty($dsID)) {
        drupal_set_message(t('You must specify an object pid and datastream ID.'), 'error');
        return '';
    }
    if (!fedora_repository_access(ObjectHelper :: $OBJECT_HELPER_VIEW_FEDORA, $pid, $user)) {
        drupal_set_message(t('You do not have the appropriate permissions'), 'error');
        return;
    }

    module_load_include('inc', 'fedora_repository', 'api/fedora_item');
    $item = new fedora_item($pid);
    $datastream_list = $item->get_datastreams_list_as_SimpleXML();
    if (!isset($datastream_list)) {
        //drupal_set_message( t("No datastreams available."), 'status' );
        return '';
    }
    foreach ($datastream_list as $datastream) {
        foreach ($datastream as $datastreamValue) {
            if ($datastreamValue->ID == $dsID) {
                return $datastreamValue->MIMETYPE;
            }
        }
    }
    return '';
}

/**
 * getDatastreamInfo ??
 * @global type $user
 * @param type $pid
 * @param type $dsID
 * @return type
 */
function getDatastreamInfo($pid, $dsID) {
    global $user;
    if (empty($pid) || empty($dsID)) {
        drupal_set_message(t('You must specify an object pid and datastream ID.'), 'error');
        return '';
    }
    if (!fedora_repository_access(ObjectHelper :: $OBJECT_HELPER_VIEW_FEDORA, $pid, $user)) {
        drupal_set_message(t('You do not have the appropriate permissions'), 'error');
        return;
    }

    module_load_include('inc', 'fedora_repository', 'api/fedora_item');
    $item = new fedora_item($pid);
    $datastream_list = $item->get_datastreams_list_as_SimpleXML();
    if (!isset($datastream_list)) {
        //drupal_set_message( t("No datastreams available."), 'status' );
        return '';
    }
    foreach ($datastream_list as $datastream) {
        foreach ($datastream as $datastreamValue) {
            if ($datastreamValue->ID == $dsID) {
                return $datastreamValue;
            }
        }
    }
    return '';
}

```

```

/**
 * internal function
 * @param $pid String
 * @param $dataStreamValue Object
 */
function create_link_for_ds($pid, $dataStreamValue) {
    global $base_url;
    $path = drupal_get_path('module', 'fedora_repository');

    require_once($path . '/api/fedora_item.inc');
    $item = new Fedora_Item($pid);

    if (user_access(ObjectHelper :: $PURGE_FEDORA_OBJECTSANDSTREAMS)) {
        $allow = TRUE;
        if (module_exists('fedora_fesl')) {
            $allow = fedora_fesl_check_roles($pid, 'write');
        }
        if ($allow) {
            $purgeImage = '<a title="purge datastream" . $dataStreamValue->label . "' href="' . $base_url .
            '/fedora/repository/purgeStream/' .
            $pid . '/' . $dataStreamValue->ID . '/' . $dataStreamValue->label . "'></a>';
        }
    }
    else {
        $purgeImage = '&nbsp;';
    }
    $fullPath = base_path() . $path;

    // Add an icon to replace a datastream
    // @TODO Note: using l(theme_image(..), ..); for these image links (and other links) may remove the need to
    have clean urls enabled.
    $replaceImage = '&nbsp;';
    if (user_access(ObjectHelper :: $ADD_FEDORA_STREAMS)) {
        $allow = TRUE;
        if (module_exists('fedora_fesl')) {
            $allow = fedora_fesl_check_roles($pid, 'write');
        }
        if ($allow) {
            $replaceImage = '<a title="' . t("Replace datastream") . "' . $dataStreamValue->label . "' href="' .
            $base_url . '/fedora/repository/replaceStream/' . $pid . '/' . $dataStreamValue->ID . '/' . $dataStreamValue-
            >label . "'></a>';
        }
    }

    $content = '';
    $id = $dataStreamValue->ID;
    $label = $dataStreamValue->label;
    $label = str_replace("_", " ", $label);
    $mimeType = $dataStreamValue->MIMETYPE;

    $view = '<a href="' . $base_url . '/fedora/repository/' . drupal_urlencode($pid) . '/' . $id . '/' .
    drupal_urlencode($label) .
    "' target='_blank' >' . t('View') . '</a>';
    $action = "$base_url/fedora/repository/object_download/" . drupal_urlencode($pid) . '/' . $id . '/' .
    drupal_urlencode(preg_replace('/\\/\\/i', '{$1}_', $label)); // Necessary to handle the case of Datastream labels
    that contain slashes. Ugh.
    $downloadVersion = '<form method="GET" action="' . $action . "'><input type="submit" value="' . t('Download')
    . "'></form>';
    if (user_access(ObjectHelper :: $EDIT_FEDORA_METADATA)) {
        $versions = $item->get_datastream_history($id);
        if (is_array($versions)) {
            $downloadVersion = '<form method="GET" action="' . $action . "' onsubmit="this.action=\"' . $action . '\"
            + '\\/' . $this.version.value;";
            $downloadVersion .= '<select name="version">';
            foreach ($versions as $version) {
                $downloadVersion .= '<option>' . $version->createDate . '</option>';
            }
            $downloadVersion .= '</select><input type="submit" value="' . t('Download') . "'></form>';
        }
    }

    $content .= "<tr><td>$label</td><td>&nbsp;$view</td><td>&nbsp;$downloadVersion</td><td>&nbsp;";
    $mimeType</td><td>&nbsp;$replaceImage&nbsp;$purgeImage</td></tr>>\n";
    return $content;
}

/**
 * getFormattedDC ??
 * @global type $base_url
 * @param type $item
 * @return type
 */
function getFormattedDC($item) {
    global $base_url;
    $path = drupal_get_path('module', 'fedora_repository');
    module_load_include('inc', 'fedora_repository', 'ConnectionHelper');

    $dsid = array_key_exists('QDC', $item->get_datastreams_list_as_array()) ? 'QDC' : 'DC';
    $xmlstr = $item->get_datastream_dissemination($dsid);

    if (empty($xmlstr)) {
        return '';
    }
}

```

```

try {
  $proc = new XsltProcessor();
} catch (Exception $e) {
  drupal_set_message($e->getMessage(), 'error');
  return;
}

$proc->setParameter('', 'baseUrl', $base_url);
$proc->setParameter('', 'path', $base_url . '/' . $path);
$input = NULL;
$xml = new DomDocument();
try {
  $xml->load($path . '/xsl/convertQDC.xsl');
  $input = new DomDocument();
  $input->loadXML(trim($xmlstr));
} catch (Exception $e) {
  watchdog(t("Fedora_Repository"), "Problem loading XSL file: !e", array('!e' => $e), NULL, WATCHDOG_ERROR);
}
$xml = $proc->importStylesheet($xml);
$newdom = $proc->transformToDoc($input);
$output = $newdom->saveHTML();
return $output;
}

/**
 * Queries fedora for what we call the qualified dublin core. Currently only dc.coverage has
 * any qualified fields
 * Transforms the returned xml to html
 * This is the default metadata view. With icons for searching a dublin core field
 * @param $pid String
 * @return String
 */
function getQDC($pid) {
  global $base_url;
  $item = new Fedora_Item($pid);
  $ds_list = $item->get_datastreams_list_as_array();
  $output = $this->getFormattedDC($item);
  $dsid = array_key_exists('QDC', $ds_list) ? 'QDC' : 'DC';
  $path = drupal_get_path('module', 'fedora_repository');

  // $baseUrl = substr($baseUrl, 0, (strpos($baseUrl, '/') - 1));
  if (user_access(ObjectHelper::EDIT_FEDORA_METADATA)) {
    $allow = TRUE;
    if (module_exists('fedora_fesl')) {
      $allow = fedora_fesl_check_roles($pid, 'write');
    }
    if ($allow) {
      $output .= '<br /><a title = "' . t('Edit Meta Data') . '" href="' . $base_url . '/fedora/repository/' .
        'editmetadata/' . $pid . '/' .
        $dsid . '"></a>';
    }
  }
  return $output;
}

/**
 * Gets a list of datastreams from an object using its pid
 *
 * We make some assumptions here. We have implemented a policy that
 * we ingest in our repository will have TN (thumbnail) datastream. Even audio
 * will have a picture of a speaker or something. This is not critical
 * but makes searches etc. look better if there is a TN stream.
 * This displays all the streams in a collapsed fieldset at the bottom of the object page.
 * you can implement a content model if you would like certain streams displayed in certain ways.
 * @param $object_pid String
 * @return String
 */
function get_formatted_datastream_list($object_pid, $contentModels, &$fedoraItem) {
  global $fedoraUser, $fedoraPass, $base_url, $user;
  module_load_include('inc', 'fedora_repository', 'ConnectionHelper');
  module_load_include('inc', 'fedora_repository', 'ObjectHelper');
  module_load_include('inc', 'fedora_repository', 'api/fedora_item');
  module_load_include('inc', 'fedora_repository', 'ContentModel');

  $path = drupal_get_path('module', 'fedora_repository');
  $dataStreamBody = '';
  $fedoraItem = new Fedora_Item($object_pid);

  if (user_access(ObjectHelper::VIEW_DETAILED_CONTENT_LIST)) {
    $availableDataStreamsText = 'Detailed List of Content';
    // $metaDataText = 'Description';
    $mainStreamLabel = NULL;
    $object = $fedoraItem->get_datastreams_list_as_SimpleXML();
    if (!isset($object)) {
      drupal_set_message(t("No datastreams available"));
      return '';
    }
    $hasOBJStream = NULL;
    $hasTNStream = FALSE;
    $dataStreamBody = "<br /><table>\n";

    $cmDataStreams = array();

```

```

    if (variable_get('fedora_object_restrict_datastreams', FALSE) == TRUE && ($cm =
ContentModel::loadFromObject($object_pid) != FALSE) {
    $cmDatastreams = $cm->listDatastreams();
    }

    $dataStreamBody .= $this->get_parent_objects_asHTML($object_pid);
    $dataStreamBody .= '<tr><th colspan="4"><h3>' . t('!text', array('!text' => $availableDataStreamsText)) .
'</h3></th></tr>';
    foreach ($object as $datastream) {
    foreach ($datastream as $datastreamValue) {
    if (variable_get('fedora_object_restrict_datastreams', FALSE) == FALSE || ((isset($user) &&
in_array('administrator', $user->roles)) || in_array($datastreamValue->ID, $cmDatastreams))) {
    if ($datastreamValue->ID == 'OBJ') {
    $hasOBJStream = '1';
    $mainStreamLabel = $datastreamValue->label;
    $mainStreamLabel = str_replace("_", " ", $mainStreamLabel);
    }
    if ($datastreamValue->ID == 'TN') {
    $hasTNStream = TRUE;
    }
    //create the links to each datastream
    $dataStreamBody .= $this->create_link_for_ds($object_pid, $datastreamValue); //"<tr><td><b>$key
:</b></td><td>$value</td></tr>\n";
    }
    }
    }
    $dataStreamBody .= "</table>\n";
    //if they have access let them add a datastream
    if (user_access(ObjectHelper::ADD_FEDORA_STREAMS)) {
    $allow = TRUE;
    if (module_exists('fedora_fesl')) {
    $allow = fedora_fesl_check_roles($object_pid, 'write');
    }
    if ($allow) {
    $dataStreamBody .= drupal_get_form('add_stream_form', $object_pid);
    }
    }
    $fieldset = array(
    '#title' => t('!text', array('!text' => $availableDataStreamsText)),
    '#collapsible' => TRUE,
    '#collapsed' => TRUE,
    '#value' => $dataStreamBody
    );
    $dataStreamBody = '<div class = "fedora_detailed_list">' . theme('fieldset', $fieldset) . '</div>';

    return $dataStreamBody;
    }
    return '';
    }

/**
 * returns a stream from a fedora object given a pid and dsid
 */
function getStream($pid, $dsid, $showError = FALSE) {
    module_load_include('inc', 'fedora_repository', 'ConnectionHelper');
    $soapHelper = new ConnectionHelper();
    try {
    $client = $soapHelper->getSoapClient(variable_get('fedora_soap_url',
'http://localhost:8080/fedora/services/access?wsdl'));
    $params = array(
    'pid' => "$pid",
    'dsID' => "$dsid",
    'asOfDateTime' => ""
    );

    if (!isset($client)) {
    drupal_set_message(t('Error connection to Fedora using soap client.'));
    return NULL;
    }
    $object = $client->__soapCall('getDatastreamDissemination', array('parameters' => $params));
    } catch (Exception $e) {
    if ($showError) {
    drupal_set_message(t('Error getting Datastream for %pid and %datastream<br />', array('%pid' => $pid,
'%datastream' => $dsid)), 'error');
    }
    return NULL;
    }
    $content = $object->dissemination->stream;
    $content = trim($content);
    return $content;
    }

/**
 * gets the name of the content models for the specified object
 * this now returns an array of pids as in Fedora 3 we can have more then one Cmodel for an object
 * @param type $pid
 * @param type $include_fedora_system_content_models
 * @return array
 */
function get_content_models_list($pid, $include_fedora_system_content_models = FALSE) {
    module_load_include('inc', 'fedora_repository', 'CollectionClass');
    $collectionHelper = new CollectionClass();
    $pids = array();
    $query = 'select $object from <#ri>
    where <info:fedora/' . $pid . '> <fedora-model:hasModel> $object

```

```

        and $object <fedora-model:state> <info:fedora/fedora-system:def/model#Active>;
$content_models = $collectionHelper->getRelatedItems($pid, $query);

if (empty($content_models)) {
    return $pids;
}

try {
    $sxml = new SimpleXMLElement($content_models);
} catch (exception $e) {
    watchdog(t("Fedora_Repository"), "Could not find a parent object for %s", $pid, NULL, WATCHDOG_ERROR);
    return $pids;
}

if (!isset($sxml)) {
    return $pids;
}
}

$content_models = array();
foreach ($sxml->xpath('//@uri') as $uri) {
    if (strpos($uri, 'fedora-system:FedoraObject-3.0') != FALSE && $include_fedora_system_content_models ==
FALSE) {
        continue;
    }
    $cmodel_pid = substr(strstr($uri, '/'), 1);
    $cm = ContentModel::loadFromModel($cmodel_pid);
    if ($cm) {
        $content_models[] = $cm;
    }
}

return $content_models;
}

/**
 * determines whether we can see the object or not
 * checks PID namespace permissions, and user permissions
 * @global type $user
 * @param type $op
 * @param type $pid
 * @return type
 */
function fedora_repository_access($op, $pid) {
    global $user;

    $returnValue = FALSE;
    $isRestricted = variable_get('fedora_namespace_restriction_enforced', TRUE);
    if (!$isRestricted) {
        $namespaceAccess = TRUE;
    }
    if ($pid == NULL) {
        $pid = variable_get('fedora_repository_pid', 'islandora:root');
    }
    $namespaceAllowed = explode(" ", variable_get('fedora_pids_allowed', 'default: demo: changeme: islandora:
ilives: islandora-book: books: newspapers: '));
    $pos = NULL;
    foreach ($namespaceAllowed as $namespace) {
        $pos = strpos($pid, $namespace);
        if ($pos === 0) {
            $namespaceAccess = TRUE;
        }
    }
    if ($namespaceAccess) {
        $user_access = user_access($op);
        if ($user_access == NULL) {
            return FALSE;
        }
        return $user_access;
    }
    else {
        return FALSE;
    }
}

/**
 * internal function
 * uses an xsl to parse the sparql xml returned from the ITQL query
 *
 *
 * @param $content String
 */
function parseContent($content, $pid, $dsId, $collection, $pageNumber = NULL) {
    $path = drupal_get_path('module', 'fedora_repository');
    global $base_url;
    $collection_pid = $pid; //we will be changing the pid later maybe
    //module_load_include('php', 'Fedora_Repository', 'ObjectHelper');
    $objectHelper = $this;
    $parsedContent = NULL;
    $contentModels = $objectHelper->get_content_models_list($pid);
    $isCollection = FALSE;
    //if this is a collection object store the $pid in the session as it will come in handy
    //after a purge or ingest to return to the correct collection.

    $fedoraItem = NULL;
    $datastreams = $this->get_formatted_datastream_list($pid, $contentModels, $fedoraItem);

    if (!empty($contentModels)) {

```

```

foreach ($contentModels as $contentModel) {
    if ($contentModel == variable_get('fedora_collection_model_pid', 'islandora:collectionCModel')) {
        $SESSION['fedora_collection'] = $pid;
        $isCollection = TRUE;
    }
}

if ($fedoraItem != NULL) {
    $dslist = $fedoraItem->get_datastreams_list_as_array();
    if (isset($dslist['COLLECTION_POLICY'])) {
        $isCollection = TRUE;
    }
}

//-----
//show the collections datastreams
if ($results->length > 0 || $isCollection == TRUE) {
    // if(strlen($objectList)>22||$contentModel=='Collection'||$contentModel=='Community')//length of empty
    dom still equals 22 because of <table/> etc
    module_load_include('inc', 'Fedora_Repository', 'CollectionPolicy');
    $collectionPolicyExists = $objectHelper->getMimeType($pid, CollectionPolicy::getDefaultDSID());
    if (user_access(ObjectHelper::: $INGEST_FEDORA_OBJECTS) && $collectionPolicyExists) {
        if (!empty($collectionPolicyExists)) {
            $allow = TRUE;
            if (module_exists('fedora_fesl')) {
                $allow = fedora_fesl_check_roles($pid, 'write');
            }
            if ($allow) {
                // $ingestObject = '<a title="' . t('Ingest a New object into ') . $collectionName . ' ' .
                $collection_pid . '" href="' . $base_path() .
                $ingestObject = '<a title="' . t('Ingest a New object into !collection name PID !collection_pid',
                array('!collection_name' => $collectionName, '!collection_pid' => $collection_pid)) . '" href="' . $base_path()
                .
                'fedora/ingestObject/' . $collection_pid . '/' . $collectionName . '"></a> ' . t('Add to this
                Collection');
            }
        }
        else {
            $ingestObject = '&nbsp;';
        }
    }
}

$datastreams .= $ingestObject;

$output .= $datastreams;

$showDesc = FALSE;
switch (variable_get('fedora_object_display_description', ObjectHelper::: $DISPLAY_NO_MODEL_OUTPUT)) {
    case ObjectHelper::: $DISPLAY_NEVER: break;
    case ObjectHelper::: $DISPLAY_NO_MODEL_OUTPUT:
        if (trim($datastreams) == '') {
            $showDesc = TRUE;
        }
        break;
    case ObjectHelper::: $DISPLAY_ALWAYS:
    default:
        $showDesc = TRUE;
        break;
}

if ($showDesc) {
    //just show default dc or qdc as we could not find a content model
    $metaDataText = t('Description');
    $body = $this->getQDC($pid);
    $fieldset = array(
        '#title' => t("!" . $metaDataText, array('!metaDataText' => $metaDataText)),
        '#collapsible' => TRUE,
        '#collapsed' => TRUE,
        '#value' => $body
    );
    $output .= theme('fieldset', $fieldset);
}

return $output;
}

/**
 * Gets the parent objects that this object is related to
 *
 * @param unknown_type $pid
 * @return unknown
 */
function get_parent_objects($pid) {
    $query_string = 'select $object $title from <#ri>
        where ($object <dc:title> $title
            and <info:fedora/' . $pid . '> <fedora-reis-ext:isMemberOfCollection> $object
            and $object <fedora-model:state> <info:fedora/fedora-system:def/model#Active>
            order by $title';
    $objects = $this->getCollectionInfo($pid, $query_string);
    return $objects;
}

```

```

/**
 * get_parent_objects_asHTML ??
 * @global type $base_url
 * @param type $pid
 * @return string
 */
function get_parent_objects_asHTML($pid) {
    global $base_url;
    $parent_collections = $this->get_parent_objects($pid);
    try {
        $parent_collections = new SimpleXMLElement($parent_collections);
    } catch (exception $e) {
        drupal_set_message(t('Error getting parent objects !e', array('!e' => $e->getMessage())));
        return;
    }

    $parent_collections_HTML = '';
    foreach ($parent_collections->results->result as $result) {
        $collection_label = $result->title;
        foreach ($result->object->attributes() as $a => $b) {
            if ($a == 'uri') {
                $uri = (string) $b;
                $uri = $base_url . '/fedora/repository' . substr($uri, strpos($uri, '/')) . '/-' . $collection_label;
            }
            $parent_collections_HTML .= '<a href="' . $uri . '"> . $collection_label . '</a><br />';
        }
        if (!empty($parent_collections_HTML)) {
            $parent_collections_HTML = '<tr><td><h3>' . t("Belongs to these collections:") . ' </h3></td><td colspan="4">' . $parent_collections_HTML . '</td></tr>';
        }

        return $parent_collections_HTML;
    }

/**
 * gets a list of datastreams and related function that we should use to show datastreams in their own fieldsets
 * from the content model associated with the object
 *
 * @param type $pid
 * @param type $contentModel
 * @param type $page_number
 * @return type
 */
function createExtraFieldsets($pid, $contentModel, $page_number) {
    // $models = $collectionHelper->getContentModels($collectionPid, FALSE);
    // possible problem in below if the collection policy has multiple content models
    // with different pids but same dsid we could get wrong one, low probability and functionality
    // will probably change for fedora version 3.
    if (empty($contentModel)) {
        return NULL;
    }
    $output = '';
    module_load_include('inc', 'fedora_repository', 'ContentModel');
    if (($cm = ContentModel::loadFromModel($contentModel)) !== FALSE && $cm->validate()) {
        $output .= $cm->displayExtraFieldset($pid, $page_number);
    }
    return $output;
}

/**
 * Look in the content model for rules to run on the specified datastream.
 *
 * @param string $pid
 * @param string $dsid
 * @return boolean
 */
function get_and_do_datastream_rules($pid, $dsid, $file = '') {
    if (!user_access('ingest new fedora objects')) {
        drupal_set_message(t('You do not have permission to add datastreams.'));
        return FALSE;
    }

    module_load_include('inc', 'fedora_repository', 'ContentModel');
    if ($dsid != NULL && $pid != NULL && ($cm = ContentModel::loadFromObject($pid)) !== FALSE) {
        $cm->execAddDatastreamMethods($dsid, $file);
    }
}

/**
 * Get a tree of related pids - for the basket functionality
 *
 * @param type $pid
 * @return type
 */
function get_all_related_pids($pid) {
    if (!$pid) {
        return FALSE;
    }
    module_load_include('inc', 'fedora_repository', 'api/fedora_utils');

    // Get title and descriptions for $pid
    $query_string = 'select $title $desc from <#ri>
        where $o <dc:title> $title

```

```

        and $o <dc:description> $desc
        and $o <mulgara:is> <info:fedora/' . $pid . '>';

$url = variable_get('fedora_repository_url', 'http://localhost:8080/fedora/risearch');
$url .= "?type=tuples&flush=true&format=csv&limit=1000&lang=itql&stream=on&query=";
$content = do_curl($url . htmlentities(urlencode($query_string)));

$rows = explode("\n", $content);
$fields = explode(',', $rows[1]);

$pids[$pid] = array('title' => $fields[0], 'description' => $fields[1]);
// $pids += $this->get_child_pids(array($pid));
}
return $pids;
}

/**
 * Get children of PID - but only 2 levels deep
 *
 * @param type $pids
 * @return type
 */
function get_child_pids($pids) {
    // Get pid, title and description for children of object $pid
    $query_string = 'select $o $title from <#ri> ' .
// $query_string = 'select $o $title $desc from <#ri> ' .
    'where $s <info:fedora/fedora-system:def/relations-external#hasMember> $o ' .
    'and $o <dc:title> $title ' .
// 'and $o <dc:description> $desc ' .
    'and ( ' ;

    foreach ($pids as $pid) {
        $query_string .= '$s <mulgara:is> <info:fedora/' . $pid . '> or ' ;
    }
    $query_string = substr($query_string, 0, -3) . ')';

    $url = variable_get('fedora_repository_url', 'http://localhost:8080/fedora/risearch');
    $url .= "?type=tuples&flush=true&format=csv&limit=1000&lang=itql&stream=on&query=";
    $url .= htmlentities(urlencode($query_string));
    $content = $this->doCurl($url);

    $rows = explode("\n", $content);
    // Knock of the first heading row
    array_shift($rows);

    $child_pids = array();
    if (count($rows)) {
        // iterate through each row
        foreach ($rows as $row) {
            if ($row == "") {
                continue;
            }
            $fields = explode(',', $row);
            $child_pid = substr($fields[0], 12);
            $child_pids[$child_pid] = array('title' => $fields[1], 'description' => $fields[2]);
        }
        if (!empty($child_pids)) {
            $child_pids += $this->get_child_pids(array_keys($child_pids));
        }
    }

    return $child_pids;
}

/**
 * Returns XML description of the object (export).
 *
 * @param type $pid
 * @param type $context
 * @param type $format
 * @return type
 */
function getObject($pid, $context = 'archive', $format = FOXML_11) {
    module_load_include('inc', 'fedora_repository', 'api/fedora_utils');

    $url = variable_get('fedora_base_url', 'http://localhost:8080/fedora') . '/objects/' . $pid . '/export?
context=' . $context . '&format=' . $format;
    $result_data = do_curl($url);
    return $result_data;
}

/**
 * Builds an array of drupal links for use in breadcrumbs.
 *
 * @global type $base_url
 * @param type $pid
 * @param type $breadcrumbs
 * @param type $level
 */
function getBreadcrumbs($pid, &$breadcrumbs, $level=10) {
    module_load_include('inc', 'fedora_repository', 'api/fedora_utils');
    // Before executing the query, we hve a base case of accessing the top-level collection
    global $base_url;
    if ($pid == variable_get('fedora_repository_pid', 'islandora:root')) {
        $breadcrumbs[] = l(t('Digital repository'), 'fedora/repository');
    }
}

```

```

    $breadcrumbs[] = l(t('Home'), $base_url);
  }
  else {
    $query_string = 'select $parentObject $title $content from <#ri>
      where (<info:fedora/' . $pid . '> <dc:title> $title
      and $parentObject <fedora-model:hasModel> $content
      and (<info:fedora/' . $pid . '> <fedora-rels-ext:isMemberOfCollection>

$parentObject
      or <info:fedora/' . $pid . '> <fedora-rels-ext:isMemberOf> $parentObject
      or <info:fedora/' . $pid . '> <fedora-rels-ext:isPartOf> $parentObject)
      and $parentObject <fedora-model:state> <info:fedora/fedora-

system:def/model#Active>
      minus $content <mulgara:is> <info:fedora/fedora-system:FedoraObject-3.0>
      order by $parentObject';

    $query_string = htmlentities(urlencode($query_string));

    $url = variable_get('fedora_repository_url', 'http://localhost:8080/fedora/risearch');
    $url .= "?type=Tuples&flush=TRUE&format=CSV&limit=1&offset=0&lang=itql&stream=on&query=" . $query_string;

    $result = preg_split('/[\r\n]+/', do_curl($url));
    array_shift($result); // throw away first line
    $matches = str_getcsv(implode("\n", $result));
    if ($matches !== FALSE) {
      $parent = preg_replace('/^info:fedora\/', '', $matches[0]);
      if ((strlen($parent) > 0) && ($level < 10)) {
        if (substr($matches[1], -11, 6) != '- page') {
          if (strlen($matches[1]) > 25) {
            $breadcrumbs[] = l(htmlentities(substr($matches[1], 0, 25)) . '...', 'fedora/repository/' . $pid);
          }
          else {
            $breadcrumbs[] = l($matches[1], 'fedora/repository/' . $pid);
          }
        }
      }
    }

    if ($parent == variable_get('fedora_repository_pid', 'islandora:root')) {
      $breadcrumbs[] = l(t('Digital repository'), 'fedora/repository');
      $breadcrumbs[] = l(t('Home'), $base_url);
    }
    elseif ($level > 0) {
      $this->getBreadcrumbs($parent, $breadcrumbs, $level - 1);
    }
  }
}

/**
 * warnIfMisconfigured ??
 * @param type $app
 */
public static function warnIfMisconfigured($app) {
  $messMap = array(
    'Kakadu' => 'Full installation instructions for Kakadu can be found
      <a href=http://www.kakadusoftware.com/index.php?option=com_content&task=view&id=27&Itemid=23>Here</a>',
    'ImageMagick' => 'Check the path settings in the configuration of your <b>imageapi</b> module.<br/>
      Further details can be found <a href=http://www.imagemagick.org/script/install-source.php>Here</a>',
  );

  $warnMess = "Creation of one or more datastreams failed.<br width='40%' align = 'left'/>";
  $configMess = "Please ensure that %app is installed and configured for this site. ";
  drupal_set_message($warnMess, 'warning', FALSE);
  drupal_set_message(t($configMess . "<br />" . $messMap[$app]) . "<br width='40%' align = 'left'/>",
  array('%app' => $app), 'warning', FALSE);
}
}

```

6.1.2 sites/all/themes/digibess/digibess.css

```

a:link, a:visited {
    color: #0000ce;
}
#wrapper #container #header h1, #wrapper #container #header h1 a:link, #wrapper #container #header h1 a:visited {
    color: #0000ce;
    font-weight: normal;
    text-shadow: #0000ce 0px 0px 0px;
    font-size: 1.7em;
}
#header-region {
    min-height: 1em;
    background: #fff;
}
#wrapper {
    background: #fff;
}
#wrapper #container #center #squeeze {
    background: #fff url(images/bg-content.png) repeat-x 50% 0;
    position: relative;
}
#wrapper #container .sidebar .block {
    margin-right: 10px;
    padding-left: 5px;
}
#block-islandora_solr_search-advanced {
    background: #fff url(images/bg-content.png) repeat-x 50% 0;
}
#block-islandora_solr_search-advanced h2 {
    /* background: #fff url(images/bg-tabs.png) repeat-x 50% 0; */
    font-size: 1.3em;
    /* height: 24px; */
    padding: 16px 0 0 0;
    color: #000;
}
#block-islandora_solr_search-basic_facets h2 {
    font-size: 1.3em;
    color: #000;
    background: #fff url(images/bg-tabs.png) repeat-x 50% 0;
}
#block-islandora_solr_search-basic_facets h3 {
    font-size: 1em;
    color: #000;
    background: #fff url(images/bg-tabs.png) repeat-x 50% 0;
}
#tabs-tabset {
    background: #fff;
}
#tabs-tabset ul.primary li a {
    background: #eeef6 url(images/bg-tabs.png) repeat-x 50% 0;
    color: #0000ce;
}
#tabs-tabset ul.primary li a:hover {
    color: #000;
}
#tabs-tabset ul.primary li.ui-tabs-selected a {
    color: #000;
}
#tabs-tabset .collview {
    background: transparent;
}
#tabs-tabset .collview td {
    padding: 1px 1px 0px 1px;
    text-align: center;
}
#tabs-tabset .collcell1 {
    /* background: #fff url(images/bg-content.png) repeat-x 50% 0; */
    /* border: 5px solid #fff; */
}
#tabs-tabset .cellthumb {
    background: #fff;
    border: 1px solid #fff;
    vertical-align: bottom;
}
#tabs-tabset .cellthumb img{
    border: 1px solid #dbdbf8;
}
#tabs-tabset .cellthumb1 {
    border: 1px solid #eeef6;
    vertical-align: middle;
    font-size: 0.6em;
    height: 150px;
}
#tabs-tabset .cellthumb1 img{
    border: 0px solid #dbdbf8;
}
#tabs-tabset tbody {
    border: 0px solid #fff;
}
#tabs-tabset .libromd img{
    border: 1px solid #dbdbf8;
}
#tabs-tabset .celldescr {
    background: #fff;
    border: 1px solid #fff;
}
    
```

```

vertical-align: top;
color: #000;
}
#tabs-tabset .celldescr1 {
border: 5px solid #fff;
vertical-align: top;
height: 30px;
background: #fff url(images/bg-tabs.png) repeat-x 50% 0;
color: #000;
}
#tabs-tabset .tocnr_collection .label {
border: 3px solid #fff;
width: 10%;
height: 150px;
padding: 15px 3px 5px 5px;
text-align: center;
vertical-align: top;
}
#tabs-tabset .tocnr_collection .value {
border: 3px solid #fff;
width: 90%;
height: 80px;
background: #fff url(images/bg-content.png) repeat-x 50% 0;
vertical-align: top;
text-align: right;
padding: 15px 5px 5px 3px;
color: #000;
}
#tabs-tabset .tocnr_collection .value h3{
background: transparent;
padding: 2px 5px 5px 3px;
color: #000;
text-align: left;
}
#tabs-tabset .tocnr_collection .value1 {
vertical-align: top;
}
#tabs-tabset .tocnr_collection .value1 a, a:link, a:visited {
color: #0000ce;
}
#tabs-tabset .tocnr_collection .value1 a:hover {
text-decoration: none;
}
#tabs-tabset h2 {
color: #000;
}
#tabs-tabset h3, h4 {
background: #fff url(images/bg-tabs.png) repeat-x 50% 0;
color: #000;
}
#wrapper #container #center .right-corner {
background: transparent;
position: relative;
left: 10px;
}
#wrapper #container #center .right-corner .left-corner {
padding: 60px 25px 5em 35px;
background: transparent;
margin-left: -10px;
position: relative;
left: -10px;
min-height: 400px;
}
#wrapper #container .breadcrumb {
color: #909090;
}
#wrapper #container .breadcrumb a {
color: #0000ce;
}
ul.primary-links {
margin: 4.5em 0 0 0;
padding: 0;
float: right; /* LTR */
position: relative;
z-index: 4;
background: #fff;
}
ul.primary-links li {
margin: 0;
padding: 0;
float: left; /* LTR */
background-image: none;
}
ul.primary-links li a, ul.primary-links li a:link, ul.primary-links li a:visited {
display: block;
margin: 0 2px;
padding: 2px 5px 0px 5px;
color: #0000ce;
/* background: transparent url(images/bg-navigation-item.png) no-repeat 50% 0; */
background: #fff url(images/bg-tabs.png) repeat-x 50% 0;
height: 24px;
width: 100px;
text-align: center;
}
ul.primary-links li a.active {
color: #000;
}

```

```

/* background: transparent url(images/bg-navigation-item-hover.png) no-repeat 50% 0; */
}
ul.primary-links li a:hover {
/* background: transparent url(images/bg-navigation-item-hover.png) no-repeat 50% 0; */
text-decoration: none;
}

```

6.1.3 sites/all/modules/islandora-github/CollectionClass.inc

```

<?php
/**
 * @file
 *
 * Collection Class Class
 */
/**
 * This CLASS caches the streams so once you call a getstream once it will always return
 * the same stream as long as you are using the instance of this class. Cached to
 * prevent mutiple hits to fedora. maybe a bit confusing though if this class is used in
 * a different context.
 */
class CollectionClass {

    public static $COLLECTION_CLASS_COLLECTION_POLICY_STREAM = 'COLLECTION_POLICY';
    public static $COLLECTION_CLASS_COLLECTION_VIEW_STREAM = 'COLLECTION_VIEW';
    private $contentModelStream = NULL;
    private $collectionPolicyStream = NULL;
    private $collectionViewStream = NULL;
    public $collectionObject = NULL;

    /**
     * Creates a collection object. Optionally can associate it with a single collection with parameter $pid.
     *
     * @param string $pid The pid of the collection to represent.
     * @return CollectionClass
     */
    function __construct($pid = NULL) {
        if (!empty($pid)) {
            module_load_include('inc', 'fedora_repository', 'ObjectHelper');
            $this->collectionObject = new ObjectHelper($pid);
            $this->pid = $pid;
        }
    }

    /**
     * gets objects related to this object. must include offset and limit
     * calls getRelatedItems but enforces limit and offset
     * @param type $pid
     * @param type $limit
     * @param type $offset
     * @param type $itqlquery
     * @return type
     */
    function getRelatedObjects($pid, $limit, $offset, $itqlquery=NULL) {
        module_load_include('inc', 'fedora_repository', 'ObjectHelper');
        $objectHelper = new ObjectHelper();
        if (!isset($itqlquery)) {
            // $query_string = $objectHelper->getStream($pid, 'QUERY', 0);
            $itqlquery = $objectHelper->getStream($pid, 'QUERY', 0);
        }
        return $this->getRelatedItems($pid, $itqlquery, $limit, $offset);
    }

    /**
     * Gets objects related to this item. It will query the object for a Query stream and use that as a itql query
     * or if there is no query stream it will use the default. If you pass a query to this method it will use the
     * passed in query no matter what
     * @global type $user
     * @param type $pid
     * @param type $itqlquery
     * @param int $limit
     * @param int $offset
     * @return type
     */
    function getRelatedItems($pid, $itqlquery = NULL, $limit = NULL, $offset = NULL) {
        module_load_include('inc', 'fedora_repository', 'ObjectHelper');
        module_load_include('inc', 'fedora_repository', 'api/fedora_utils');
        if (!isset($limit)) {
            $limit = 1000;
        }
        if (!isset($offset)) {
            $offset = 0;
        }
        global $user;
        if (!fedora_repository_access(OBJECTHELPER :: $OBJECT_HELPER_VIEW_FEDORA, $pid, $user)) {
            drupal_set_message(t("You do not have access to Fedora objects within the attempted namespace or access to
            Fedora denied."), 'error');
            return ' ';
        }
        $objectHelper = new ObjectHelper();
        $query_string = $itqlquery;
        if (!isset($query_string)) {
            $query_string = $objectHelper->getStream($pid, 'QUERY', 0);
        }
    }
}

```

```

    if ($query_string == NULL) {
        $query_string = 'select $object $title $content from <#ri>
            where ($object <dc:title> $title
                and $object <fedora-model:hasModel> $content
                and ($object <fedora-rels-ext:isMemberOfCollection> <info:fedora/' . $pid . '>
                    or $object <fedora-rels-ext:isMemberOf> <info:fedora/' . $pid . '>))
                and $object <fedora-model:state> <info:fedora/fedora-system:def/model#Active>
                minus $content <mulgara:is> <info:fedora/fedora-system:FedoraObject-3.0>
                order by $title';
    }
}
else {
    // Replace $parent_collection% with the actual collection PID
    $query_string = preg_replace("/\%parent_collection%/", "<info:fedora/$pid>", $query_string);
}

$query_string = htmlentities(urlencode($query_string));

$content = '';
$url = variable_get('fedora_repository_url', 'http://localhost:8080/fedora/research');
$url .= "?type=tuples&flush=TRUE&format=Sparql&limit=$limit&offset=$offset&lang=itql&stream=on&query=" .
$query_string;
$content .= do_curl($url);

return $content;
}

/**
 * get Collection Policy Stream ?
 * @param type $collection_pid
 * @return type
 */
function getCollectionPolicyStream($collection_pid) {
    if ($this->collectionPolicyStream != NULL) {
        return $this->collectionPolicyStream;
    }
    $this->collectionPolicyStream = $this->getStream($collection_pid, CollectionClass ::
COLLECTION_CLASS_COLLECTION_POLICY_STREAM, 0);
    return $this->collectionPolicyStream;
}

/**
 * get Relationship element ?
 * @param type $collection_pid
 * @return type
 */
function getRelationshipElement($collection_pid) {
    $stream = $this->getCollectionPolicyStream($collection_pid);
    try {
        $xml = new SimpleXMLElement($stream);
    } catch (Exception $e) {
        drupal_set_message(t('Error getting relationship element from policy stream !e', array('!e' => $e-
>getMessage()))), 'error');
        return;
    }
    $relationship = $xml->relationship[0];
    return $relationship;
}

/**
 * get Collection View Stream ?
 * @param type $collection_pid
 * @return type
 */
function getCollectionViewStream($collection_pid) {
    $this->collectionViewStream = $this->getStream($collection_pid, CollectionClass ::
COLLECTION_CLASS_COLLECTION_VIEW_STREAM, 0);
    return $this->collectionViewStream;
}

/**
 * get Stream ?
 * @param type $pid
 * @param type $dsid
 * @param type $showError
 * @return type
 */
function getStream($pid, $dsid, $showError = 1) {
    module_load_include('inc', 'fedora_repository', 'api/fedora_item');
    $item = new fedora_item($pid);
    return isset($item->datastreams[$dsid]) ? $item->get_datastream_dissemination($dsid) : NULL;
}

/**
 * get Pid name space ?
 * @param type $pid
 * @param type $dsid
 * @return type
 */
function getPidNameSpace($pid, $dsid) {
    $stream = $this->getCollectionPolicyStream($pid);
    try {
        $xml = new SimpleXMLElement($stream);
    } catch (Exception $e) {
        drupal_set_message(t('Error getting PID namespace !e', array('!e' => $e->getMessage())), 'error');
        return;
    }
}

```

```

    }
    foreach ($xml->contentmodels->contentmodel as $contentModel) {
        // $contentModelName=strip_tags($contentModel['name']);
        $contentModelDsid = strip_tags($contentModel->dsid);
        if (strtolower($contentModelDsid) == strtolower($dsid)) {
            return strip_tags($contentModel->pid_namespace->asXML());
        }
    }
    drupal_set_message(t('Error getting PID namespace! using collection pid of !pid and content model of !dsid',
array('!pid' => $pid, '!dsid' => $dsid)), 'error');
    return NULL;
}

/**
 * gets a list of content models from a collection policy
 * @param type $collection_pid
 * @param type $showError
 * @return ContentModel
 */
function getContentModels($collection_pid, $showError = TRUE) {
    module_load_include('inc', 'Fedora_Repository', 'ContentModel');
    $collection_stream = $this->getCollectionPolicyStream($collection_pid);
    try {
        $xml = new SimpleXMLElement($collection_stream);
    } catch (Exception $e) {
        if ($showError) {
            drupal_set_message(t("!e", array('!e' => $e->getMessage())), 'error');
        }
        return NULL;
    }
    foreach ($xml->contentmodels->contentmodel as $content_model) {
        $contentModel = new ContentModel();
        $contentModel->contentModelDsid = $content_model->dsid;
        $contentModel->contentModelPid = $content_model->pid;
        $contentModel->pidNamespace = $content_model->pidNamespace;
        $contentModel->contentModelName = $content_model['name'];
        $models[] = $contentModel;
    }
    return $models;
}

/**
 * using the collection policies pid namespace get a new pid by calling fedora' get next pid and appending it
to the namespace
 * $pid is the $pid of the content model
 * $dsid is the datastream id of the content model.
 */
function getNextPid($pid, $dsid) {
    module_load_include('inc', 'Fedora_Repository', 'ConnectionHelper');
    $pidNameSpace = $this->getPidNameSpace($pid, $dsid);
    $pname = substr($pidNameSpace, 0, strpos($pidNameSpace, ":"));
    module_load_include('inc', 'fedora_repository', 'api/fedora_item');
    return Fedora_Item::get_next_pid_in_namespace($pname);
}

/**
 * gets the form handler file, class and method and returns them in an array
 *
 * @param string $pid The content model PID
 * @param string $dsid The content model DSID
 * @return array The file, class and method name to handle the ingest form.
 */
function getFormHandler($pid, $dsid) {
    $stream = $this->getContentModelStream($pid, $dsid);
    try {
        $xml = new SimpleXMLElement($stream);
    } catch (Exception $e) {
        drupal_set_message(t('Error Getting FormHandler: !e', array('!e' => $e->getMessage())), 'error');
        return NULL;
    }
    $formHandler = $xml->ingest_form;
    if ($formHandler != NULL) {
        $handlerDsid = strip_tags($formHandler['dsid']);
        $handlerMethod = strip_tags($formHandler->form_builder_method->form_handler->asXML());
        $handlerFile = strip_tags($formHandler->form_builder_method->file->asXML());
        $handlerClass = strip_tags($formHandler->form_builder_method->class_name->asXML());
        $handlerModule = strip_tags($formHandler->form_builder_method->module->asXML());
        $returnArray = array();
        $returnArray['module'] = $handlerModule;
        $returnArray['method'] = $handlerMethod;
        $returnArray['class'] = $handlerClass;
        $returnArray['file'] = $handlerFile;
        return $returnArray;
    }
}

drupal_set_message(t("Error getting form handler. No handler found for !pid - !dsid", array('!pid' => $pid,
'!dsid' => $dsid)), 'error');
return NULL;
}

/**
 * get Allowed Mime Types
 * @param type $contentModelPid
 * @param type $contentModelDsid
 * @return type
 */

```

```

function getAllowedMimeTypes($contentModelPid, $contentModel_dsid) {
    $stream = $this->getContentModelStream($contentModelPid, $contentModel_dsid);
    try {
        $xml = new SimpleXMLElement($stream);
    } catch (Exception $e) {
        drupal_set_message(t('Error getting content model stream for mime types !e', array('!e' => $e->getMessage())), 'error');
        return;
    }
    foreach ($xml->mimetypes->type as $type) {
        $types[] = $type;
    }
    return $types;
}

/**
 * Grabs the rules from the content model stream
 * file the file that has been uploaded
 *
 * @param type $file
 * @param type $mimetype
 * @param type $pid
 * @param type $dsid
 * @return type
 */
function getAndDoRules($file, $mimetype, $pid, $dsid) {
    if (!user_access('ingest new fedora objects')) {
        drupal_set_message(t('You do not have permission to ingest objects.'));
        return FALSE;
    }

    $stream = $this->getContentModelStream($pid, $dsid);

    try {
        $xml = new SimpleXMLElement($stream);
    } catch (Exception $e) {
        drupal_set_message(t('Error getting content model stream !e', array('!e' => $e->getMessage())), 'error');
        return FALSE;
    }
    foreach ($xml->ingest_rules->rule as $rule) {
        foreach ($rule->applies_to as $type) {
            if (!strcmp(trim($type), trim($mimetype))) {
                $methods = $rule->methods->method;
                return $this->callMethods($file, $methods);
            }
        }
    }
}

/**
 * calls the methods defined in the content model rules .xml file stored in a Fedora object
 *
 * @param type $file
 * @param type $methods
 * @return type
 */
function callMethods($file, $methods) {
    foreach ($methods as $method) {
        $phpFile = strip_tags($method->file->asXML());
        $phpClass = strip_tags($method->class_name->asXML());
        $phpMethod = strip_tags($method->method_name->asXML());
        $file_ext = strip_tags($method->modified_files_ext->asXML());
        $parameters = $method->parameters->parameter;
        $dsid = strip_tags($method->datastream_id);
        $parametersArray = array();
        if (isset($parameters)) {
            foreach ($parameters as $parameter) {
                $name = strip_tags($parameter->name);
                $value = strip_tags($parameter->asXML());
                $parametersArray["$name"] = $value;
            }
        }
        // module_load_include($phpFile, 'Fedora_Repository', ' ');
        require_once(drupal_get_path('module', 'fedora_repository') . '/' . $phpFile);
        $thisClass = new $phpClass ();
        $returnValue = $thisClass->$phpMethod($parametersArray, $dsid, $file, $file_ext);
        if (!$returnValue) {
            drupal_set_message('Error! Failed running content model method !m !rv', array('!m' => $phpMethod, '!rv' => $returnValue));
            return FALSE;
        }
    }
    return TRUE;
}

/**
 * grabs a xml form definition from a content model and builds
 * the form using drupals forms api
 *
 * @param type $form
 * @param type $form_state
 * @param type $contentModelPid
 * @param type $contentModelDsid
 * @return type
 */
function build_ingest_form(&$form, &$form_state, $contentModelPid, $contentModelDsid) {

```

```

$stream = $this->getContentModelStream($contentModelPid, $contentModelDsid);
try {
    $xml = new SimpleXMLElement($stream);
} catch (Exception $e) {
    drupal_set_message(t('Error getting ingest form stream !e', array('!e' => $e->getMessage())), 'error');
    return FALSE;
}
$docRoot = $_SERVER['DOCUMENT_ROOT'];

$file = (isset($form_state['values']['ingest-file-location']) ? $form_state['values']['ingest-file-location']
: '');
// $fullPath = $docRoot . ' . $file;
$fullpath = $file;
// $form = array();
$form['step'] = array(
    '#type' => 'hidden',
    '#value' => (isset($form_state['values']['step']) ? $form_state['values']['step'] : 0) + 1,
);
$form['ingest-file-location'] = array(
    '#type' => 'hidden',
    '#value' => $file,
);

$form['content_model_name'] = array(
    '#type' => 'hidden',
    '#value' => $contentModelDsid
);
$form['models'] = array(//content models available
    '#type' => 'hidden',
    '#value' => $form_state['values']['models'],
);
$form['fullpath'] = array(
    '#type' => 'hidden',
    '#value' => $fullpath,
);

$form['#attributes']['enctype'] = 'multipart/form-data';
$form['indicator']['ingest-file-location'] = array(
    '#type' => 'file',
    '#title' => t('Upload Document'),
    '#size' => 48,
    '#description' => t('Full text'),
);
if ($xml->ingest_form->hide_file_chooser == 'TRUE') {
    $form['indicator']['ingest-file-location']['#type'] = 'hidden';
}
$ingest_form = $xml->ingest_form; //should only be one
$drupal_module = strip_tags($ingest_form->form_builder_method->module->asXML());
if (empty($drupal_module)) {
    $drupal_module = 'fedora_repository';
}
$phpFile = strip_tags($ingest_form->form_builder_method->file->asXML());
$phpClass = strip_tags($ingest_form->form_builder_method->class_name->asXML());
$phpMethod = strip_tags($ingest_form->form_builder_method->method_name->asXML());
$dsid = strip_tags($ingest_form['dsid']);
// module_load_include('php', 'Fedora_Repository', $phpFile);
require_once(drupal_get_path('module', $drupal_module) . '/' . $phpFile);
$thisClass = new $phpClass ();

return $thisClass->$phpMethod($form, $ingest_form, $form_state['values'], $form_state);
}

/**
 * this will also create a personal collection for an existing user if they don't have one
 * not using this function currently
 *
 * @param type $user
 * @return type
 */
function createUserCollection(& $user) {
    if (isset($user->fedora_personal_pid)) {
        return;
    }
    $username = array(
        'name' => variable_get('fedora_admin_user', 'fedoraAdmin')
    );
    $admin_user = user_load($username);
    module_load_include('inc', 'fedora_repository', 'ConnectionHelper');
    $connectionHelper = new ConnectionHelper();
    try {
        $soapClient = $connectionHelper->getSoapClient(variable_get('fedora_soap_manage_url',
        'http://localhost:8080/fedora/services/management?wsdl'));
        $pidNameSpace = variable_get('fedora_repository_pid', 'vre:');
        $pidNameSpace = substr($pidNameSpace, 0, strpos($pidNameSpace, ":"));
        $params = array(
            'numPIDs' => '',
            'pidNameSpace' => $pidNameSpace
        );
        $object = $soapClient->__soapCall('getNextPID', array(
            $params
        ));
    } catch (exception $e) {
        drupal_set_message(t('Error getting Next PID: !e', array('!e' => $e->getMessage())), 'error');
        return FALSE;
    }
    $pid = implode(get_object_vars($object));
}

```

```

$pidNumber = strstr($pid, ":");
$pid = $pidNameSpace . ':' . 'USER-' . $user->name . '-' . substr($pidNumber, 1);
$personal_collection_pid = array(
  'fedora_personal_pid' => $pid
);
module_load_include('inc', 'fedora_repository', 'plugins/PersonalCollectionClass');
$personalCollectionClass = new PersonalCollectionClass();
if (!$personalCollectionClass->createCollection($user, $pid, $soapClient)) {
  drupal_set_message("Did not create a personal collection object for !u", array('!u' => $user->name));
  return FALSE; //creation failed don't save the collection pid in drupal db
}
user_save($user, $personal_collection_pid);
return TRUE;
}

/**
 * Queries a collection object for an xslt to format how the
 * collection of objects is displayed.
 *
 * @param type $pid
 * @param type $path
 * @param type $canUseDefault
 * @return type
 */
function getXslContent($pid, $path, $canUseDefault = TRUE) {
  module_load_include('inc', 'fedora_repository', 'CollectionClass');
  $collectionClass = new CollectionClass();
  $xslContent = $collectionClass->getCollectionViewStream($pid);
  if (!$xslContent && $canUseDefault) { //no xslt so we will use the default sent with the module
    $xslContent = file_get_contents($path . '/xsl/sparql_to_html.xsl');
  }
  return $xslContent;
}

/**
 * show field sets ?
 * @global type $base_url
 * @global type $user
 * @param type $page_number
 * @return string
 */
function showFieldSets($page_number) {
  module_load_include('inc', 'fedora_repository', 'api/fedora_item');
  global $base_url;
  $tabset = array();
  global $user;
  $objectHelper = new ObjectHelper();
  $item = new Fedora_Item($this->pid);
  $query = NULL;
  if ($item->exists() && array_key_exists('QUERY', $item->datastreams)) {
    $query = $item->get_datastream_dissemination('QUERY');
  }
  $results = $this->getRelatedItems($this->pid, $query);

  $colleinfo = NULL;
  if ($item->exists() && array_key_exists('INFO', $item->datastreams)) {
    $colleinfo = $item->get_datastream_dissemination('INFO');
  }

  $collection_items = $this->renderCollection($results, $this->pid, NULL, NULL, $page_number);
  $collection_item = new Fedora_Item($this->pid);
  // Check the form post to see if we are in the middle of an ingest operation.
  $show_ingest_tab = (!empty($_POST['form_id']) && $_POST['form_id'] == 'fedora_repository_ingest_form');
  $add_to_collection = $this->getIngestInterface();

  $view_selected = true;
  $coll_selected = false;

  $qstring = $_GET['q'];
  $qparts = explode('/', $qstring);
  $tail = end($qparts);

  if ($tail == 'info') {
    $view_selected = false;
    $coll_selected = true;
  }

  drupal_set_message();

  $tabset['1'] = array(
    '#type' => 'tabpanel',
    '#title' => $collection_item->objectProfile->objLabel,
    '#selected' => $view_selected,
    '#content' => $collection_items,
    '#weight' => '1'
  );

  if ($colleinfo != NULL) {
    $tabset['2'] = array(
      '#type' => 'tabpanel',
      '#title' => 'Collection info',
      '#selected' => $coll_selected,
      '#content' => $colleinfo,
      '#weight' => '2'
    );
  }
}

```

```

    }

    // $tabset['add_tab'] = array(
    //     #type and #title are the minimum requirements.
    //     '#type' => 'tabpanel',
    //     '#title' => t('Add'),
    //     '#selected' => $show_ingest_tab,
    //     // This will be the content of the tab.
    //     '#content' => $add_to_collection,
    // );

    // $tabset['add_collection_tab'] = array(
    //     #type and #title are the minimum requirements.
    //     '#type' => 'tabpanel',
    //     '#title' => t('Add Child Collection'),
    //     // This will be the content of the tab.
    //     '#content' => drupal_get_form('collection_creation_form', $this->pid),
    // );

    return $tabset;
}

/**
 * get Ingest Interface ??
 * @global type $base_url
 * @return string
 */
function getIngestInterface() {
    global $base_url;
    $objectHelper = new ObjectHelper();
    module_load_include('inc', 'Fedora_Repository', 'CollectionPolicy');
    $collectionPolicyExists = $objectHelper->getMimeType($this->pid, CollectionPolicy::getDefaultDSID());
    if (user_access(ObjectHelper::INGEST_FEDORA_OBJECTS) && $collectionPolicyExists) {
        if (!empty($collectionPolicyExists)) {
            $allow = TRUE;
            if (module_exists('fedora_fesl')) {
                $allow = fedora_fesl_check_roles($this->pid, 'write');
            }
            if ($allow) {
                $ingestObject = drupal_get_form('fedora_repository_ingest_form', $this->pid);
            }
        }
    }
    else {
        $ingestObject = '';
    }

    return $ingestObject;
}

/**
 * render collection
 * @global type $base_url
 * @param type $content
 * @param type $pid
 * @param type $dsId
 * @param type $collection
 * @param int $pageNumber
 * @return type
 */
function renderCollection($content, $pid, $dsId, $collection, $pageNumber = NULL) {
    $path = drupal_get_path('module', 'fedora_repository');
    global $base_url;
    $collection_pid = $pid; //we will be changing the pid later maybe
    $objectHelper = new ObjectHelper();
    $parsedContent = NULL;
    $contentModels = $objectHelper->get_content_models_list($pid);
    $isCollection = FALSE;
    //if this is a collection object store the $pid in the session as it will come in handy
    //after a purge or ingest to return to the correct collection.

    $fedoraItem = NULL;

    $collectionName = $collection;
    if (!$pageNumber) {
        $pageNumber = 1;
    }

    if (!isset($collectionName)) {
        $collectionName = variable_get('fedora_repository_name', 'Collection');
    }
    $xslContent = $this->getXslContent($pid, $path);

    //get collection list and display using xslt-----
    $objectList = '';
    if (isset($content) && $content != FALSE) {
        $input = new DomDocument();
        $input->loadXML(trim($content));
        $results = $input->getElementsByTagName('result');
        if ($results->length > 0) {
            try {
                $proc = new XsltProcessor();
                $proc->setParameter('', 'collectionPid', $collection_pid);
                $proc->setParameter('', 'collectionTitle', $collectionName);
            }
        }
    }
}

```



```

        '#type' => 'checkboxes',
        '#options' => $content_models,
        '#required' => true,
        '#description' => t("Content models describe the behaviours of objects with which they are associated."),
    );
}

$form['submit'] = array(
    '#type' => 'submit',
    '#value' => t('Submit')
);

return($form);
}

/**
 * collection creation form validate ?
 * @param type $form
 * @param type $form_state
 * @return type
 */
function collection_creation_form_validate($form, &$form_state) {
    module_load_include('inc', 'fedora_repository', 'api/fedora_item');
    module_load_include('inc', 'fedora_repository', 'api/fedora_utils');
    $pid = $form_state['values']['collection_pid'];
    $item = new fedora_item($pid);
    if ($item->exists()) {
        form_set_error('', t("$pid already exists within your repository. the PID must be unique"));
        return;
    }
    if (!valid_pid($pid)) {
        form_set_error('', t("$pid is not a valid identifier"));
        return;
    }
}

/**
 * collection creation form submit
 * @global type $user
 * @param type $form
 * @param type $form_state
 */
function collection_creation_form_submit($form, &$form_state) {
    global $user;
    module_load_include('inc', 'fedora_repository', 'api/fedora_item');
    module_load_include('inc', 'fedora_repository', 'api/dublin_core');
    $module_path = drupal_get_path('module', 'fedora_repository');
    $thumbnail = drupal_get_path('module', 'Fedora_Repository') .
        '/images/Crystal_Clear_filesystem_folder_grey.png';
    $collection_pid = $form_state['values']['collection_pid'];
    $new_collection_label = $form_state['values']['collection_name'];
    $parent_collection = $form_state['values']['parent_collection'];
    $pid_namespace = $form_state['values']['collection_namespace'];
    $all_cModels = get_content_models_as_option_array();
    $collection_policy = '<?xml version="1.0" encoding="UTF-8"?>
    <collection_policy xmlns="http://www.islandora.ca" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="''
    xsi:schemaLocation="http://www.islandora.ca http://syn.lib.umanitoba.ca/collection_policy.xsd">
        <content_models>
        </content_models>
        <search_terms>
        </search_terms>
        <staging_area></staging_area>
        <relationship>isMemberOfCollection</relationship>
    </collection_policy>';
    $content_models = $form_state['values']['content_models'];
    $collection_policy_xml = simplexml_load_string($collection_policy);
    foreach ($content_models as $content_model) {
        if ($content_model) {
            $node = $collection_policy_xml->content_models->addChild('content_model');
            $node->addAttribute('dsid', 'ISLANDORACM');
            $node->addAttribute('name', $all_cModels[$content_model]);
            $node->addAttribute('namespace', $pid_namespace . ':1');
            $node->addAttribute('pid', $content_model);
        }
    }
    $item = fedora_item::ingest_new_item($collection_pid, 'A', $new_collection_label, $user->name);
    $item->add_relationship('isMemberOfCollection', $parent_collection, RELS_EXT_URI);
    $item->add_relationship('hasModel', 'islandora:collectionCModel', FEDORA_MODEL_URI);
    $item->add_datastream_from_string($collection_policy_xml->saveXML(), 'COLLECTION_POLICY');
    $item->add_datastream_from_file($thumbnail, 'TN');
    drupal_goto("/fedora/Repository/$collection_pid");
}

```

6.1.4 sites/all/modules/islandora_solr_custom/theme/islandora_solr_custom.theme.inc

```

<?php
/**
 * @file
 * This file holds all of islandora_solr_custom's theme functions
 */
/**
 * Theme function for theming the admin table
 * @return rendered table
 */
function theme_islandora_solr_custom_admin_settings($form)
{
  $rows = array();
  foreach ($form['islandora_solr_custom_table'] as $key)
  {
    $row = array();
    // check if $key is really the array we need. we shouldn't select the #parent array for example.
    if (isset($key['field']['#type']))
    {
      $row[] = array(
        'data' => drupal_render($key['field'])
      );
      $row[] = array(
        'data' => drupal_render($key['label'])
      );
      $row[] = array(
        'data' => drupal_render($key['exclude_label'])
      );
      $row[] = array(
        'data' => drupal_render($key['markup'])
      );
      $rows[] = $row;
    }
  }
  $form['islandora_solr_custom_table']['#access'] = FALSE; // don't unset $form['anything'] , use
  $form['anything']['#access'] = FALSE; instead!
  // Individual table headers.
  $header = array();
  $header[] = t('solr field');
  $header[] = t('label');
  $header[] = t('exclude label?');
  $header[] = t('show markup when empty?');
  // put the rendered table in another fieldset.
  $form['islandora_solr_custom_table_output']['table_output_markup']['#value'] = theme('table', $header, $rows);
  $output = '';
  $output .= drupal_render($form);
  return $output;
}
/**
 * Theme function for theming the solr search results
 *
 * @param $results
 * The raw search results
 *
 * @return variables to be used in the template file.
 */
function islandora_solr_custom_preprocess_islandora_solr_custom(&$variables)
{
  // base url
  global $base_url;
  $variables['base_url'] = $base_url;
  // include solr field order generated from solrconfig.xml
  // $variables['field_order'] = _islandora_solr_custom_solrconfig(); // not sure if this is really needed, but
  // I'll keep it in for now.
  // this variable might be more useful as the db values are parsed in the solrconfig order using the results of
  // islandora_solr_custom_solrconfig().
  // get the solrconfig.xml and DB results. Don't check or fix any differences between solrconfig.xml and the DB
  $variables['db_values'] = _islandora_solr_custom_db_values($check = FALSE);
  $db_values = $variables['db_values'];
  $variables['display'] = variable_get('islandora_solr_custom_display', 'div');
  $variables['display_url'] = (isset($_GET['display']) ? $_GET['display'] : NULL);
  $variables['table_rendered'] = ''; // set the variable. If the table style is selected, this variable will be
  // populated with a rendered table.
  $variables['switch'] = variable_get('islandora_solr_custom_switch', 0);
  $variables['switch_rendered'] = ''; // set the variable. If the display switch is enabled, this variable will
  // be populated with a rendered link.
  $variables['css'] = variable_get('islandora_solr_custom_css', 1);
  // set $variables['style'];
  $variables['style'] = '';
  if ($variables['display_url'] != NULL)
  {
    if ($variables['display_url'] == 'div')
    {
      $variables['style'] = 'div';
    }
    elseif ($variables['display_url'] == 'table')
    {
      $variables['style'] = 'table';
    }
    else
    {
      $variables['style'] = $variables['display'];
    }
  }
  else

```

```

    {
        $variables['style'] = $variables['display'];
    }
    // include css or not
    $css = variable_get('islandora_solr_custom_css', 1);
    if ($css == 1)
    {
        drupal_add_css(drupal_get_path('module', 'islandora_solr_custom') . '/css/islandora_solr_custom_tpl.css');
    }
    /***/
    // $pippo = drupal_get_breadcrumb();
    // $briciola = $pippo[sizeof($pippo)-1];
    // $primo = strpos($briciola, "\"", 0)+1;
    // $lunga = strpos($briciola, "\"", $primo+1)-$primo;
    // $briciola = substr($briciola, $primo, $lunga);
    // $fulltxt = strpos($briciola, "OCR.OCR");
    /***/
    $hl = $variables['results_raw']->highlighting;
    // go over each search result object
    foreach ($variables['results_raw']->response->docs as $doc)
    {
        $resultsArray = array();
        // go over each field within a search result object
        foreach ($db_values as $db_value)
        {
            $field = $db_value['field'];
            /***/
            // if ($field == 'rels.hasModel') {
            //     $modello = $doc->$db_value['field'];
            // }
            /***/
            if (isset($doc->$db_value['field']))
            {
                $value = $doc->$db_value['field'];
                if (is_array($value))
                {
                    $value = implode(" ", $value);
                }
                if ($field != "OCR.OCR")
                {
                    $resultsArray[$field]['value'] = $value;
                }
            }
            else
            {
                $resultsArray[$field]['value'] = '';
            }
            if ($field == "PID")
            {
                $h1v = $hl->$value;
                foreach ($h1v as $singlef)
                {
                    $linee = "";
                    foreach ($singlef as $linea)
                    {
                        $linee = $linee . "... " . $linea . "... <br>";
                    }
                    $resultsArray['OCR.OCR']['value'] = $linee;
                }
            }
            if (($field == "dc.description") && ($value != ""))
            {
                $resultsArray[$field]['value'] = substr($value, 0, 200) . "... ";
            }
            // set the fields
            $resultsArray[$field]['label'] = $db_value['label'];
            $resultsArray[$field]['class'] = strtolower(preg_replace('/[^\A-Za-z0-9]/', '-', $db_value['field'])); //
            // turn the field name in a string that is appropriate to use as a class name.
            $resultsArray[$field]['exclude_label'] = $db_value['exclude_label'];
            $resultsArray[$field]['markup'] = $db_value['markup'];
        }
        /***/
        //if ($fulltxt == "") {
        // if ($modello == "info:fedora/islandora:bookCModel") {
        $variables['results'][] = $resultsArray;
        // }
        // }
        /***/
    }
    // if 'display' is set to table, render the table.
    if ($variables['style'] == 'table')
    {
        $db_values = $variables['db_values'];
        $results = $variables['results'];
        $output = theme('islandora_solr_custom_table', $db_values, $results);
        $variables['table_rendered'] = $output;
    }
    // if display switch is set, render the link
    if ($variables['switch'] == 1 AND $variables['results'] != '')
    {
        $display = ''; // set $display
        $db_display = $variables['display'];
        $switch_style = ''; // this is will be the oposite display style than then one enabled.
        if ($db_display == 'table')
        {
            $switch_style = 'div';
        }
    }

```

```

    }
    elseif ($db_display == 'div')
    {
        $switch_style = 'table';
    }
    if (isset($_GET['display']))
    {
        if ($_GET['display'] == 'table')
        {
            $display = 'div';
        }
        elseif ($_GET['display'] == 'div')
        {
            $display = 'table';
        }
        else
        {
            $display = $switch_style; // if the display style is not properly set in the url, choose the opposite of
            the one enabled in the DB.
        }
    }
    else
    {
        $display = $switch_style; // if the display style is not set in the url, choose the opposite of the one
        enabled in the DB.
    }
}
$curr_url = $base_url . request_uri();
if (isset($_GET['page']))
{
    $curr_url = str_replace('&display=' . $_GET['display'], '', $curr_url);
    $curr_url = $curr_url . '&display=' . $display;
}
else
{
    $curr_url = str_replace('?display=' . $_GET['display'], '', $curr_url);
    $curr_url = $curr_url . '?display=' . $display;
}
$variables['switch_rendered'] = theme('islandora_solr_custom_switch', $curr_url);
}
// $variables['template_file'] = 'islandora-solr-custom-test';
}
/**
 * Theme function for theming the switch link.
 * The theme function is really minimal, but allows for overriding they way the switch is themed.
 *
 * @return rendered switch link
 */
function theme_islandora_solr_custom_switch($curr_url)
{
    return t('Switch display', $curr_url);
}
/**
 * Theme function for theming the table display.
 *
 * @return rendered table display
 */
function theme_islandora_solr_custom_table($db_values, $results)
{
    $output = '';
    if ($results == '')
    {
        $output = '<p> . t('Your search yielded no results') . '</p>';
    }
    else
    {
        // Table headers.
        $header = array();
        foreach ($db_values as $key => $result)
        {
            if ($result['exclude_label'] == 0)
            {
                $header[] = array(
                    'data' => t($result['label'])
                    //'field' => $result['field'], // I guess sorting is only possible when the result is fetched from the
                    DB.
                );
            }
            else
            {
                $header[] = '';
            }
        }
        // Table rows
        $rows = array();
        foreach ($results as $key => $result)
        {
            $row = array();
            foreach ($result as $key => $value)
            {
                $row[] = array(
                    'data' => $value['value']
                );
            }
            $rows[] = $row;
        }
        $output = theme('table', $header, $rows);
    }
}

```

```

    }
    return $output;
}
/**
 * Theme function for setting the values of the <item> elements for the RSS secondary display.
 *
 * @tutorial http://feed2.w3.org/docs/rss2.html#hrelementsOfLitemgt
 * @return variable that holds all values to be rendered into <item> elements
 */
function theme_islandora_solr_custom_rss_item($doc, $feed)
{
    // set the variables to be rendered as elements in the <item> element
    $result['title'] = $doc['dc.title'];
    $result['link'] = $base_url . '/fedora/repository/' . htmlspecialchars($doc['PID'], ENT_QUOTES, 'utf-8');
    $result['description'] = $doc['dc.description'];
    $result['items'] = array(
        array(
            'key' => 'author',
            'value' => $doc['dc.creator']
        ),
        array(
            'key' => 'guid',
            'value' => $doc['PID'],
            'attributes' => array(
                'isPermaLink' => 'false'
            )
        ),
        array(
            'key' => 'pubDate',
            'value' => $doc['figs.createdDate']
        ),
        array(
            'key' => 'category',
            'value' => $doc['dc.subject']
        ),
        array(
            'key' => 'enclosure',
            'value' => '',
            'attributes' => array(
                'url' => '',
                'length' => '',
                'type' => ''
            )
        ),
        array(
            'key' => 'comments',
            'value' => ''
        ),
        array(
            'key' => 'source',
            'value' => ''
        )
    );
    return $result;
}
/**
 * Theme function for setting the values of the <channel> elements for the RSS secondary display.
 *
 * @tutorial http://feed2.w3.org/docs/rss2.html#requiredChannelElements
 * @return variable that holds all values to be rendered into <channel> elements
 */
function theme_islandora_solr_custom_rss_channel($query, $feed)
{
    global $base_url;
    // set the variables to be rendered as elements in the <channel> element
    $result['title'] = t('@site_name aggregator', array(
        '@site_name' => variable_get('site_name', 'Drupal')
    ));
    $result['url'] = $base_url;
    $result['description'] = t('Aggregated search results of:') . ' ' . $query;
    $result['langcode'] = NULL;
    $result['args'] = array();
    return $result;
}

```

6.1.5 sites/all/modules/islandora_solr_custom/theme/islandora-solr-custom.tpl.php

```

<?php
/**
 * @file islandora-solr-custom.tpl.php
 * Islandora solr search results template
 *
 * Variables available:
 * - $variables: all array elements of $variables can be used as a variable. e.g. $base_url equals
 * $variables['base_url']
 * - $base_url: The base url of the current website. eg: http://example.com .
 * - $user: The user object.
 *
 * - $style: the style of the display ('div' or 'table'). Set in admin page by default. Overridden by the query
 * value: ?display=foo
 * - $results: the array containing the solr search results
 * - $table_rendered: If the display style is set to 'table', this will contain the rendered table.
 * For theme overriding, see: theme_islandora_solr_custom_table()
 * - $switch_rendered: The rendered switch to toggle between display styles
 * For theme overriding, see: theme_islandora_solr_custom_switch()
 */

```

```

*/
?>

<?php// print $switch_rendered; ?>
<?php $style = 'table'; ?>

<?php if ($style == 'div'): ?>

<?php elseif ($style == 'table'): ?>

<?php// print $table_rendered; ?>
<?php if ($results == ''): print '<p>' . t('Your search yielded no results') . '</p>'; ?>
<?php else: ?>
  <table class="islandora_solr_results">
    <?php foreach ($results as $id => $result): ?>
      <?php $modello = $result['rels.hasModel']['value']; ?>
      <?php if ($modello == "info:fedora/islandora:pageCModel"): ?>
        <tr class="islandora_solr_result">
          <td class="label" rowspan="2">
            <?php print "<a href=\"".base_path().\"fedora/repository/\".substr($result['PID']['value'],0,-5).\"\"><img src=\"".base_path().\"fedora/repository/\".substr($result['PID']['value'], 0,-5).\"/TN\"></img></a>"; ?>
          </td>
          <td class="value">
            <?php print "<b>".substr($result['dc.title']['value'],0,-12).\"</b>"; ?><br><br>
            <i><?php print $result['dc.creator']['value']; ?></i><br>
            <i><?php print $result['dc.publisher']['value']; ?></i><br>
            <i><?php print $result['dc.description']['value']; ?></i><br>
          </td>
          <td class="plabel" rowspan="2">
            <?php print "<a href=\"".base_path().\"fedora/repository/\".$result['PID']['value'].\"\"><img height=\"150\" src=\"".base_path().\"fedora/repository/\".$result['PID']['value'].\"/TN\"></img></a>"; ?>
            <?php print "<br><b>".substr($result['dc.title']['value'],-9).\"</b>"; ?>
          </td>
        </tr>
        <tr class="islandora_solr_result">
          <td class="ocrvalue">
            <?php print $result['OCR.OCR']['value']; ?>
          </td>
        </tr>
      <?php elseif ($modello == "info:fedora/islandora:bookCModel"): ?>
        <tr class="islandora_solr_result">
          <td class="label">
            <?php print "<a href=\"".base_path().\"fedora/repository/\".$result['PID']['value'].\"\"><img src=\"".base_path().\"fedora/repository/\".$result['PID']['value'].\"/TN\"></img></a>"; ?>
          </td>
          <td class="value">
            <?php print "<b>\".$result['dc.title']['value'].\"</b>"; ?><br><br>
            <i><?php print $result['dc.creator']['value']; ?></i><br>
            <i><?php print $result['dc.publisher']['value']; ?></i><br>
            <i><?php print $result['dc.description']['value']; ?></i><br>
          </td>
        </tr>
      <?php endif; ?>
    <?php endforeach; ?>
  </table>
<?php endif; ?>
<?php endif; ?>

```

6.1.6 sites/all/modules/islandora_solr_custom/css/islandora_solr_custom_tpl.css

```

/* css markup for template */

.islandora_solr_result
{
  /* float: left; */
  margin-bottom: 25px;
  border: 10px solid #fff;
  background: #fff url(images/bg-content.png) repeat-x 50% 0;
}

.islandora_solr_result .label
{
  /* width: 15%; */
  width: 180px;
  padding: 10px 10px 0px 0px;
  vertical-align: top;
  text-align: right;
  background: #fff;
}

.islandora_solr_result .plabel
{
  /* width: 10%; */
  width: 110px;
  padding: 0px 0px 0px 5px;
  vertical-align: bottom;
  text-align: left;
}

.islandora_solr_result .value
{
  /* width: 75%; */
  padding: 10px 0px 0px 2px;
  vertical-align: top;
}

```

```

text-align: left;
/* background: #f0f0f0;*/
}

.islandora_solr_result .ocrvalue
{
/* width: 75%;*/
padding: 0px 5px 0px 5px;
vertical-align: top;
text-align: right;
font-size: 0.82em;
background: #fff;
}
    
```

6.1.7 sites/all/modules/islandora_solution_pack_book/tochr_book.inc

```

<?php
// $Id$
class IslandoraBook
{
    function __construct($pid = '')
    {
        module_load_include('inc', 'fedora_repository', 'api/fedora_item');
        if (!empty($pid))
        {
            $this->pid = $pid;
            $this->item = new Fedora_Item($pid);
        }
    }

    public function showFieldSets()
    {
        module_load_include('inc', 'fedora_repository', 'plugins/tagging_form');
        module_load_include('inc', 'fedora_repository', 'plugins/DarwinCore');
        module_load_include('module', 'islandora_book');
        global $base_url;
        $show_purge_tab = (!empty($_POST['form_id']) && ($_POST['form_id'] == 'islandora_book_purge_pages_form'));
        $first_page = new Fedora_Item($this->pid . '-0001');
        $hasPage = $first_page->exists();
        $tabset = array();
        global $user;
        $qs = '';
        if ($user->uid != 0)
        {
            $qs = '?uid=' . base64_encode($user->name . ':' . $user->pass);
        }
        $viewer_url = variable_get('fedora_base_url', '') . '/get/' . $this->pid .
        'islandora:viewerSdef/getViewer' . $qs;
        $html = '<iframe src="' . $viewer_url . '" scrolling="no" frameborder="0" style="width: 100%;
        height: 802px;">Errors: unable to load viewer</iframe>';
        $pages_selected = FALSE;
        $pagina_selected = FALSE;
        $libro_selected = FALSE;
        if ($hasPage)
        {
            $libropid = $this->pid;
            $paginapid = $this->pid . '-0001';
            $numeropagina = '0001';
            $libro_selected = TRUE;
        }
        else
        {
            $libropid = substr($this->pid, 0, -5);
            $paginapid = $this->pid;
            $numeropagina = substr($this->pid, -4);
            $pagina_selected = TRUE;
        }
        $size = 20;
        $page_number = floor(($numeropagina - 1) / $size) + 1;
        $qstring = $_GET['q'];
        $qparts = explode('/', $qstring);
        $tail = end($qparts);
        if (is_numeric($tail))
        {
            $pages_selected = TRUE;
            $pagina_selected = FALSE;
            $libro_selected = FALSE;
            $page_number = $tail;
        }
        $class = new CollectionClass();
        // grow default (1000) research limit to 2000
        $results = $class->getRelatedItems($libropid, NULL, 2000);
        $page_items = $class->renderCollection($results, $paginapid, NULL, NULL, $page_number);
        $book_contenuto = islandora_book_create_book_view($libropid);
        $titololibro = substr(strstr($book_contenuto, "<strong>title</strong></td><td>"),
        "<strong>creator</strong>", TRUE, 31);
        drupal_set_title("$titololibro");
        $stringhe = islandora_book_create_book_index($libropid);
        $indice_html = $stringhe[0];
        $colleinfo_html = $stringhe[1];
        $bookinfo_html = $stringhe[2];
        $tabset['I'] = array(
            // #type and #title are the minimum requirements.
            '#type' => 'tabpage',
            '#title' => 'BOOK',
            '#content' => $book_contenuto,
            '#selected' => $libro_selected,
            '#weight' => '1'
        );
    }
}
    
```

```

);
$tabset['2'] = array(
  '#type' => 'tabpanel',
  '#title' => t('Read'),
  '#content' => $html,
  '#weight' => '2'
);
$tabset['3'] = array(
  // #type and #title are the minimum requirements.
  '#type' => 'tabpanel',
  '#title' => 'Index',
  // '#content' => islandora_book_create_book_index($libropid),
  '#content' => $indice_html,
  '#weight' => '3'
);
$tabset['4'] = array(
  '#type' => 'tabpanel',
  '#title' => t('Thumbnails'),
  '#selected' => $pages_selected,
  '#content' => $page_items,
  '#weight' => '4'
);
$tabset['5'] = array(
  '#type' => 'tabpanel',
  '#title' => 'PAGE ' . substr($paginapid, -4),
  '#content' => islandora_book_create_book_view($paginapid),
  '#selected' => $pagina_selected,
  '#weight' => '5'
);
$tabset['6'] = array(
  '#type' => 'tabpanel',
  '#title' => 'Book info',
  '#content' => $boookinfo_html,
  '#weight' => '6'
);
$tabset['7'] = array(
  '#type' => 'tabpanel',
  '#title' => 'Collection info',
  '#content' => $colleinfo_html,
  '#weight' => '6'
);
return $tabset;
}
}

```

6.1.8 sites/all/modules/islandora_solution_pack_book/xsl/tocnr_book_view.xsl

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:variable name="OBJECTSPAGE">
    <xsl:value-of select="$objectsPage"/>
  </xsl:variable>
  <xsl:variable name="PID">
    <xsl:value-of select="$pid"/>
  </xsl:variable>
  <xsl:variable name="MODELLO">
    <xsl:value-of select="$modello"/>
  </xsl:variable>
  <xsl:variable name="ELELIB">
    <xsl:copy-of select="$elelib"/>
  </xsl:variable>
  <xsl:variable name="RECORDID">
    <xsl:value-of select="substring-after($PID, '&quot;:&quot;')"/>
  </xsl:variable>
  <xsl:variable name="COLLEZIONEID">
    <xsl:value-of select="substring-before($PID, '&quot;:&quot;')"/>
  </xsl:variable>
  <xsl:template match="/">
    <table class="libromd" cellpadding="3" cellspacing="2" valign="top">
      <tr>
        <td valign="top">
          <img>
            <xsl:attribute name="src"><xsl:copy-of select="$OBJECTSPAGE"/>fedora/repository/<xsl:copy-of
select="$PID"/>TN
            </xsl:attribute>
          </img>
        </td>
        <td>
          <div style="align:left">
            <table cellpadding="3" cellspacing="2" width="90%">
              <tbody>
                <tr>
                  <td colspan="3">
                    <h3>Dublin Core - MetaData </h3>
                  </td>
                </tr>
                <xsl:for-each select="/*/ *>
                  <xsl:variable name="FULLFIELD" select="name()"/>
                  <xsl:variable name="FIELD" select="substring-after(name(), ':')"/>
                  <xsl:variable name="DATA" select="text()"/>
                  <xsl:if test="$DATA != ' '">
                    <xsl:if test="$FIELD != 'source'">
                      <tr>
                        <td valign="top">
                          <strong>
                            <xsl:value-of select="substring-after(name(), ':')"/>
                          </strong>

```

```

        </td>
        <td>
            <xsl:value-of select="text()"/>
            <xsl:for-each select="*">
                <div><xsl:value-of select="substring-after(name(),':')"/> = <xsl:value-of
select="text()"/>
            </div>
        </xsl:for-each>
        </td>
    </tr>
</xsl:if>
</xsl:if>
</xsl:for-each>
</tbody>
<tr>
    <td colspan="3">
        <h3/>
    </td>
</tr>
<tr>
    <td colspan="3">
        <h3>Download</h3>
    </td>
</tr>
<xsl:if test="$MODELLO = 'info:fedora/islandora:bookCModel'">
    <tr>
        <td>
            <strong>Full PDF</strong>
        </td>
        <td>
            <a href="{OBJECTSPACE}fedora/repository/object_download/{PID}/PDF/
{$COLLEZIONEID}_{RECORDID}.pdf">PDF</a>
        </td>
    </tr>
</xsl:if>
<xsl:if test="$MODELLO = 'info:fedora/islandora:pageCModel'">
    <xsl:if test="not($COLLEZIONEID = 'asfiat')">
        <tr>
            <td>
                <strong>Hi-Res IMAGE</strong>
            </td>
            <td>
                <a href="{OBJECTSPACE}fedora/repository/object_download/{PID}/TIFF/
{$COLLEZIONEID}_{RECORDID}.tif">TIFF</a>
            </td>
        </tr>
    </xsl:if>
    <xsl:if test="$COLLEZIONEID = 'asfiat'">
        <tr>
            <td>
                <em>Only Low-Res available due to restricted rights</em>
            </td>
        </tr>
    </xsl:if>
    <tr>
        <td>
            <strong>Low-Res IMAGE</strong>
        </td>
        <td>
            <a href="{OBJECTSPACE}fedora/repository/object_download/{PID}/openbess:jpgldSdef/
{$COLLEZIONEID}_{RECORDID}.jpg">JPEG</a>
        </td>
    </tr>
    <tr>
        <td>
            <strong>TEXT</strong>
        </td>
        <td>
            <a href="{OBJECTSPACE}fedora/repository/object_download/{PID}/OCR/
{$COLLEZIONEID}_{RECORDID}.txt">TXT</a>
        </td>
    </tr>
</xsl:if>
<xsl:if test="$MODELLO = 'info:fedora/islandora:bookCModel'">
    <xsl:if test="$ELELIB != ''">
        <tr>
            <td colspan="3">
                <h3/>
            </td>
        </tr>
        <tr>
            <td colspan="3">
                <h3>External link</h3>
            </td>
        </tr>
        <tr>
            <td valign="top">
                <strong>Hard copy</strong>
            </td>
            <td valign="top">
                <xsl:value-of disable-output-escaping="yes" select="$ELELIB"/>
            </td>
        </tr>
    </xsl:if>
</xsl:if>
</table>

```

```

</div>
</td>
</tr>
</table>
</xsl:template>
</xsl:stylesheet>

```

6.1.9 sites/all/modules/islandora_solution_pack_book/islandora_book.module

```

<?php
// $Id$
function islandora_book_menu()
{
  $items = array();
  $items['fedora/book'] = array(
    'title' => t('Book view'),
    'page callback' => 'islandora_book_create_book_view',
    'type' => MENU_CALLBACK,
    'access arguments' => array(
      'view fedora collection'
    )
  );
  $items['fedora/book_viewer'] = array(
    'title' => t('Book viewer'),
    'page callback' => 'islandora_book_book_viewer',
    'type' => MENU_CALLBACK,
    'access arguments' => array(
      'view fedora collection'
    )
  );
  $items['admin/settings/islandora_book'] = array(
    'title' => 'Islandora Book Admin',
    'description' => 'Configuration Islandora Book Module',
    'page callback' => 'drupal_get_form',
    'page arguments' => array(
      'islandora_book_admin_settings'
    ),
    'access arguments' => array(
      'administer site configuration'
    ),
    'file' => 'islandora_book.admin.inc',
    'type' => MENU_NORMAL_ITEM
  );
  $items['islandora/book/ocr'] = array(
    'page callback' => 'update_ocr_div',
    'access arguments' => array(
      'view fedora collection'
    ),
    'file' => 'islandora_book.admin.inc',
    'type' => MENU_CALLBACK
  );
  return $items;
}

function islandora_book_book_viewer($pid)
{
  global $user;
  $qs = '';
  if ($user->uid != 0)
  {
    $qs = '?uid=' . base64_encode($user->name . ':' . $user->pass);
  }
  $viewer_url = variable_get('fedora_base_url', '') . '/get/' . $pid . '/islandora:viewerSdef/getViewer' . $qs;
  $html = '<iframe src="' . $viewer_url . '" frameborder="0" scrolling="no" style="width: 100%; height: 800px;">Errors: unable to load viewer</iframe>';
  $fieldset = array(
    '#title' => t('Viewer - ') . $pid,
    '#collapsible' => TRUE,
    '#collapsed' => FALSE,
    '#value' => $html
  );
  drupal_add_css(path_to_theme() . '/header-viewer.css', 'theme');
  return theme('fieldset', $fieldset);
}

function islandora_book_create_book_index($pid, $query = NULL)
{
  global $user;
  module_load_include('inc', 'fedora_repository', 'ObjectHelper');
  $path = drupal_get_path('module', 'islandora_book');
  $objectHelper = new ObjectHelper;
  $rels_xml = $objectHelper->getStream($pid, 'RELS-EXT');
  $simpleRELSxml = simplexml_load_string($rels_xml);
  $modello = $simpleRELSxml->xpath('//rdf:RDF/rdf:Description/fedora-model:hasModel/@rdf:resource');
  $parentcolle = $simpleRELSxml->xpath('//rdf:RDF/rdf:Description/fedora:isMemberOfCollection/@rdf:resource');
  $modello = $modello[0];
  $parentcolle = explode('/', $parentcolle[0]);
  $parentcolle = $parentcolle[1];
  if ($modello == 'info:fedora/islandora:bookCModel')
  {
    $index_html = $objectHelper->getStream($pid, 'INDEX');
    $ix_html = $objectHelper->getStream($pid, 'INDEX');
    $piddelbook = $pid;
    if (!$index_html)
    {
      drupal_set_message(t('Object does not exist.'), 'error');
      return '';
    }
  }
}

```

```

else
{
    $pidelbook = substr($pid, 0, -5);
    $index_html = $objectHelper->getStream($pidelbook, 'INDEX');
    $ix_html = $objectHelper->getStream($pidelbook, 'INDEX');
    if (!$index_html)
    {
        drupal_set_message(t('Object does not exist.'), 'error');
        return '';
    }
}
$str_html = "<table cellpadding=\"3\" cellspacing=\"2\" valign=\"top\" >";
$str_html = $str_html . "<tr><td valign=\"top\" >";
$str_html = $str_html . "<img src=\"\" . base_path() . "fedora/repository/" . $pid . "/TN\"></img>";
$str_html = $str_html . "</td><td><div style=\"align:left\"><table cellpadding=\"3\" cellspacing=\"2\" width=\"90%\">";
$str_html = $str_html . "<tbody><tr><td colspan=\"3\"><h3>Index</h3></td></tr>";
$selenco = explode("|", $ix_html);
$nvoci = sizeof($selenco);
$voce = explode("|", $selenco[0]);
$str_html = $str_html . "<tr><td><a href=\"\" . base_path() . "fedora/repository/" . $voce[0] . "\"><h5>\" . $voce[1] . "</h5>";
$str_html = $str_html . "</a></td><td></td></tr>";
$i = 1;
while ($i < $nvoci)
{
    $voce = explode("|", $selenco[$i]);
    $str_html = $str_html . "<tr><td><a href=\"\" . base_path() . "fedora/repository/" . $voce[0] . "\">\" . $voce[1];
    $numpa = substr($voce[0], -4);
    $str_html = $str_html . "</a></td><td>page&nbsp;\" . intval($numpa) . "</td></tr>";
    $i++;
}
$str_html = $str_html . "</tbody></table></div></td></tr></table>";
$strs[0] = $str_html;
$colleinfo_html = $objectHelper->getStream($parentcolle, 'INFO');
$strs[1] = $colleinfo_html;
$bookinfo_html = $objectHelper->getStream($pid, 'INFO');
$strs[2] = $bookinfo_html;
return $strs;
}

function islandora_book_create_book_hardcopy($pid, $modello)
{
    $selelib = '';
    if ($modello == "info:fedora/islandora:bookCModel")
    {
        $url = "http://oidomatic.comperio.it/getinfo.xml?href=http%3A%2F%2Fwww.digibess.it%2Ffedora%2Frepository%2F" . $pid;
        $k = 0;
        $oidom = new DOMDocument();
        if (@$oidom->load($url) !== false)
        {
            if (is_object($root = $oidom->getElementsByTagName('objects')))
            {
                if (is_object($root->item(0)))
                {
                    if (is_object($objs = $root->item(0)->getElementsByTagName('object')))
                    {
                        foreach ($objs as $obj)
                        {
                            $label = $obj->getElementsByTagName('label')->item(0)->nodeValue . "\n";
                            $href = $obj->getElementsByTagName('href')->item(0)->nodeValue . "\n";
                            if (($label !== '') && ($href !== ''))
                            {
                                // $selelib[$k] = array ( 'label' => $label, 'href' => $href );
                                // $selelib = array_merge($selelib, array ( $label => $href));
                                // $selelib .= '<p><a href=\"\".htmlentities($href).\" target=\"_blank\">\". $label. '</a></p>';
                                // $selelib .= '<p><a href=\"\".htmlentities($href, ENT_COMPAT, 'UTF-8').\" target=\"_blank\">\".htmlentities($label, ENT_COMPAT, 'UTF-8'). '</a></p>';
                                $selelib .= "<p><a href=\"\" . htmlentities($href, ENT_COMPAT, 'UTF-8') . \"\" target=\"_blank\">\" . htmlentities($label, ENT_COMPAT, 'UTF-8') . \"</a></p>";
                                $k++;
                            }
                        }
                    }
                }
            }
        }
        // if ( $k > 0 ) {
        //     print_r($selelib);
        // }
        // echo "END ( ".count($selelib). " )\n";
    }
    // return htmlentities ($selelib, ENT_COMPAT, 'UTF-8');
    return $selelib;
}

//loads an xslt for the main book page uses mods for most of the display. if there is a $query parameter
// it will execute it against the book.
function islandora_book_create_book_view($pid, $query = NULL)
{
    global $user;
    module_load_include('inc', 'fedora_repository', 'ObjectHelper');
    $path = drupal_get_path('module', 'islandora_book');
    $objectHelper = new ObjectHelper;
    // $xml = $objectHelper->getStream($pid, 'MODS');
    $xml = $objectHelper->getStream($pid, 'DC');
}

```

```

$dc_xml = $objectHelper->getStream($pid, 'DC');
$rels_xml = $objectHelper->getStream($pid, 'RELS-EXT');
if (!$rels_xml)
{
    drupal_set_message(t('Object does not exist.'), 'error');
    return '';
}
$simpleRELSxml = simplexml_load_string($rels_xml);
// $modello = $simpleRELSxml->xpath('///fedora-model:hasModel@rdf:resource');
// $modello = $simpleRELSxml->xpath('///rdf:RDF/rdf:Description/@rdf:about')."pippo";
$modello = $simpleRELSxml->xpath('///rdf:RDF/rdf:Description/fedora-model:hasModel@rdf:resource');
$modello = $modello[0];
if (!$dc_xml)
{
    drupal_set_message(t('Object does not exist.'), 'error');
    return '';
}
$simpleDCxml = simplexml_load_string($dc_xml);
$types = $simpleDCxml->xpath('/dc:type');
$ingested = 'false';
if (!empty($types))
{
    foreach ($types as $type)
    {
        if ($type == 'ingested')
        {
            $ingested = 'true';
        }
    }
}
if (!isset($pid))
{
    drupal_set_message(t('Error getting book view, no identifier specified.'));
    return;
}
$proc = NULL;
try
{
    $proc = new XsltProcessor();
}
catch (Exception $e)
{
    drupal_set_message(t('Error loading Book View XSLT: $e', array(
        'e' => $e->getMessage()
    )));
    return;
}
$elelib = islandora_book_create_book_hardcopy($pid, $modello);
//print_r($elelib);
//inject into xsl stylesheet
$proc->setParameter('', 'userID', $user->uid);
$proc->setParameter('', 'objectsPage', base_path());
$proc->setParameter('', 'pid', $pid);
$proc->setParameter('', 'ingested', $ingested);
$proc->setParameter('', 'modello', $modello);
$proc->setParameter('', 'elelib', $elelib);
$xml = new DomDocument();
$test = $xml->load($path . '/xsl/tocnr_book_view.xsl');
if (!isset($test))
{
    drupal_set_message(t('Error loading search results XSLT.'));
    return t('Error loading search results XSLT.');
```

```

$pid = str_replace(':', '?', $pid);
$newQuery = $newQuery . " AND PID:$pid* AND dc.type:Text";
return $newQuery;
}
/**
 * Custom form element to do our nice images.
 */
function islandora_book_elements() // Change this line
{
    $type['imagebutton'] = array(
        '#input' => TRUE,
        '#button_type' => 'submit',
        '#executes_submit_callback' => TRUE,
        '#name' => 'op',
        '#process' => array(
            'hook_imagebutton_process' => array()
        )
    );
    return $type;
}
function theme_imagebutton($element)
{
    return '<input style="background:url(\'\'); border:0px; width:10px; padding:0px,0px,0px,0px;" type="image"
    class="form-' . $element['#button_type'] . '" name="' . $element['#name'] . '" id="' . $element['#id'] . '"
    value="' . check_plain($element['#default_value']) . '" ' . drupal_attributes($element['#attributes']) . '
    src="' . $element['#image'] . '" alt="' . $element['#title'] . '" title="' . $element['#title'] . '" />\n';
}
/**
 * Implementation of hook_theme() to register how to theme image buttons.
 */
function islandora_book_theme()
{
    return array(
        'imagebutton' => array(
            'arguments' => array(
                'form' => NULL
            )
        ),
        'islandora_book_book_search_form' => array(
            'arguments' => array(
                'form' => NULL
            )
        ),
        'islandora_book_simple_book_search_form' => array(
            'arguments' => array(
                'form' => NULL
            )
        )
    );
}
/**
 * Implementation of hook_requirements().
 *
 * @return
 *   An array describing the status of the site regarding available updates.
 *   If there is no update data, only one record will be returned, indicating
 *   that the status of core can't be determined. If data is available, there
 *   will be two records: one for core, and another for all of contrib
 *   (assuming there are any contributed modules or themes enabled on the
 *   site). In addition to the fields expected by hook_requirements ('value',
 *   'severity', and optionally 'description'), this array will contain a
 *   'reason' attribute, which is an integer constant to indicate why the
 *   given status is being returned (UPDATE_NOT_SECURE, UPDATE_NOT_CURRENT, or
 *   UPDATE_UNKNOWN). This is used for generating the appropriate e-mail
 *   notification messages during update_cron(), and might be useful for other
 *   modules that invoke update_requirements() to find out if the site is up
 *   to date or not.
 *
 * @see _update_message_text()
 * @see _update_cron_notify()
 */
function islandora_book_requirements($phase)
{
    global $base_url;
    if ($phase == 'runtime')
    {
        module_load_include('inc', 'fedora_repository', 'api/fedora_utils');
        $requirements['iiv-war'] = array();
        $requirements['iiv-war']['title'] = t("Islandora OpenLayers image viewer web app");
        if (!islandora_book_viewer_available())
        {
            $requirements['iiv-war']['value'] = ("Not available");
            $requirements['iiv-war']['severity'] = REQUIREMENT_ERROR;
            $requirements['iiv-war']['description'] = t('Ensure that Fedora is running and that the <a href="@iiv-home">IIV</a> app is deployed.', array(
                '@iiv-home' => 'http://github.com/islandora/iiv'
            ));
        }
        else
        {
            $requirements['iiv-war']['value'] = ("Available");
            $requirements['iiv-war']['severity'] = REQUIREMENT_OK;
        }
    }
    return $requirements;
}

```

```

function islandora_book_viewer_available()
{
  $url = parse_url(variable_get('fedora_base_url', 'http://localhost:8080/fedora'));
  $fedora_host = ("{$url['scheme']}://{ $url['host']}" . (!empty($url['port']) ? ":{ $url['port']}/" : '/'));
  $response = drupal_http_request("$fedora_host/iiv/images/loading.gif");
  return empty($response->error);
}

function islandora_book_required_fedora_objects()
{
  $module_path = drupal_get_path('module', 'islandora_book');
  //add default mods form to formbuilder table
  $result = db_result(db_query('Select name from {xml_forms} where name = "%s"', 'Islandora Books MODS Form'));
  if (!$result)
  {
    $filename = "$module_path/xml/islandora_book_default_mods.xml";
    $form_as_string = file_get_contents($filename);
    $object = new stdClass();
    $object->name = 'Islandora Books MODS Form';
    $object->form = $form_as_string;
    $result = drupal_write_record('xml_forms', $object);
  }
  $result = db_result(db_query('Select name from {xml_forms} where name = "%s"', 'Upload Existing Books Mods'));
  if (!$result)
  {
    $filename = "$module_path/xml/mods_upload.xml";
    $form_as_string = file_get_contents($filename);
    $object = new stdClass();
    $object->name = 'Upload Existing Books Mods';
    $object->form = $form_as_string;
    $result = drupal_write_record('xml_forms', $object);
  }
  $result = db_result(db_query('Select content_model from {islandora_content_model_forms} where content_model = "%s" and form_name = "%s"', 'islandora:bookCModel', 'Islandora Books MODS Form'));
  if (!$result)
  {
    $object = new stdClass();
    $object->content_model = 'islandora:bookCModel';
    $object->form_name = 'Islandora Books MODS Form';
    $object->dsid = 'MODS';
    $object->title_field = "[titleInfo][title]";
    $object->transform = 'mods_to_dc.xsl';
    $result = drupal_write_record('islandora_content_model_forms', $object);
  }
  $result = db_result(db_query('Select content_model from {islandora_content_model_forms} where content_model = "%s" and form_name = "%s"', 'islandora:bookCModel', 'Upload Existing Books Mods'));
  if (!$result)
  {
    $object = new stdClass();
    $object->content_model = 'islandora:bookCModel';
    $object->form_name = 'Upload Existing Books Mods';
    $object->dsid = 'MODS';
    $object->title_field = "[title]";
    $object->transform = 'mods_to_dc.xsl';
    $result = drupal_write_record('islandora_content_model_forms', $object);
  }
  // array('path-to-foxml-file', 'pid', 'dsid', 'path-to-datastream-file', int dsversion, boolean required)
  // make database insertions here.
  return array(
    'islandora_book' => array(
      'module' => 'islandora_book',
      'title' => 'Islandora Books',
      'objects' => array(
        array(
          'foxml file' => "$module_path/xml/islandora_book_pageCModel.xml",
          'pid' => 'islandora:pageCModel',
          'dsid' => NULL,
          'datastream_file' => NULL,
          'dsversion' => NULL
        ),
        array(
          'foxml file' => "$module_path/xml/islandora_book_bookCModel.xml",
          'pid' => 'islandora:bookCModel',
          'dsid' => NULL,
          'datastream_file' => NULL,
          'dsversion' => NULL
        ),
        array(
          'foxml file' => "$module_path/xml/islandora_book_jp2sdef.xml",
          'pid' => 'islandora:jp2Sdef',
          'dsid' => NULL,
          'datastream_file' => NULL,
          'dsversion' => NULL
        ),
        array(
          'foxml file' => "$module_path/xml/islandora_book_viewerSdef.xml",
          'pid' => 'islandora:viewerSdef',
          'dsid' => NULL,
          'datastream_file' => NULL,
          'dsversion' => NULL
        ),
        array(
          'foxml file' => "$module_path/xml/islandora_book_viewerSdep-bookCModel.xml",
          'pid' => 'islandora:viewerSdep-bookCModel',
          'dsid' => NULL,
          'datastream_file' => NULL,
          'dsversion' => NULL
        )
      )
    )
  );
}

```

```

    ),
    array(
      'foxml_file' => "$module_path/xml/islandora_book_viewerSdep-pageCModel.xml",
      'pid' => 'islandora:viewerSdep-pageCModel',
      'dsid' => NULL,
      'datastream_file' => NULL,
      'dsversion' => NULL
    ),
    array(
      'foxml_file' => "$module_path/xml/islandora_book_tei2htmlSdef.xml",
      'pid' => 'islandora:tei2htmlSdef',
      'dsid' => NULL,
      'datastream_file' => NULL,
      'dsversion' => NULL
    ),
    array(
      'foxml_file' => "$module_path/xml/islandora_book_tei2htmlSdep-pageCModel.xml",
      'pid' => 'islandora:tei2htmlSdep-pageCModel',
      'dsid' => NULL,
      'datastream_file' => NULL,
      'dsversion' => NULL
    ),
    array(
      'foxml_file' => "$module_path/xml/islandora_book_collection.xml",
      'pid' => 'islandora:collection',
      'dsid' => NULL,
      'datastream_file' => NULL,
      'dsversion' => NULL
    ),
    array(
      'foxml_file' => "$module_path/xml/islandora_book_jp2Sdep-pageCModel.xml",
      'pid' => 'islandora:jp2Sdep-pageCModel',
      'dsid' => NULL,
      'datastream_file' => NULL,
      'dsversion' => NULL
    ),
    array(
      'foxml_file' => "$module_path/xml/islandora_book_CollectionModel.xml",
      'pid' => 'islandora:CollectionModel',
      'dsid' => NULL,
      'datastream_file' => NULL,
      'dsversion' => NULL
    ),
    array(
      'pid' => 'books:collection',
      'label' => 'Book Collection',
      'cmodel' => 'islandora:collectionCModel',
      'parent' => 'islandora:demos',
      'datastreams' => array(
        array(
          'dsid' => 'COLLECTION_POLICY',
          'datastream_file' => "$module_path/xml/book_collection_policy.xml"
        ),
        array(
          'dsid' => 'TN',
          'datastream_file' => "$module_path/images/Crystal_Clear_mimetype_man.png",
          'mimetype' => 'image/png'
        )
      )
    )
  )
);
}
function islandora_book_purge_pages_form(&$form_state, $pid, $referrer = NULL)
{
  global $base_url;
  if (!user_access('purge objects and datastreams'))
  {
    return NULL;
  }
  if ($pid == NULL)
  {
    return NULL;
  }
  $form['pid'] = array(
    '#type' => 'hidden',
    '#value' => "$pid"
  );
  if (!strstr(drupal_get_destination(), urlencode('fedora/repository')))
  {
    $form['referrer'] = array(
      '#type' => 'hidden',
      '#value' => $referrer
    );
  }
  if (!isset($form_state['storage']['confirm']))
  {
    // do your normal $form definition here
    $form['submit'] = array(
      '#type' => 'image_button',
      '#src' => drupal_get_path('module', 'fedora_repository') . '/images/purge_big.png',
      '#value' => t('Purge'),
      '#suffix' => 'Purge all pages associated with this book'
    );
  }
  return $form;
}

```

```

    }
    else
    {
        // ALSO do $form definition here. Your final submit handler (after user clicks Yes, I Confirm) will only see
        $form_state info defined here. Form you create here passed as param1 to confirm_form
        return confirm_form($form, 'Confirm Purge ALL pages?', $referrer, 'Are you sure you want to delete these
        objects? This action cannot be undone.', 'Purge Pages', 'Cancel'); //Had better luck leaving off last param
        'name'
    }
    return $form;
}
function islandora_book_purge_pages_form_submit($form, &$form_state)
{
    module_load_include('inc', 'fedora_repository', 'ConnectionHelper');
    $pid = $form_state['values']['pid'];
    if (!isset($form_state['storage']['confirm']))
    {
        $form_state['storage']['confirm'] = TRUE; // this will cause the form to be rebuilt, entering the confirm
        part of the form
        $form_state['rebuild'] = TRUE; // along with this
    }
    else
    {
        $query = <<<XML
select \Object \Title from <#ri>
where (\Object <dc:title> \Title
and \Object <fedora-rels-ext:isMemberOf> <info:fedora/$pid>
and \Object <fedora-model:state> <info:fedora/fedora-system:def/model#Active>)
order by \Title

XML;
// this is where you do your processing after they have pressed the confirm button
module_load_include('inc', 'fedora_repository', 'CollectionClass');
$collection = new CollectionClass($pid);
$contents = $collection->getRelatedItems($pid, $query);
$results_xml = simplexml_load_string($contents);
$resultsarray = array();
foreach ($results_xml->results->result as $result)
{
    $pid_to_delete = str_replace('info:fedora/', '', $result->object->attributes()->uri);
    //parse contents
    $params = array(
        "pid" => $pid_to_delete,
        "logMessage" => "Purged",
        "force" => ""
    );
    try
    {
        $soapHelper = new ConnectionHelper();
        $client = $soapHelper->getSoapClient(variable_get('fedora_soap_manage_url',
'http://localhost:8080/fedora/services/management?wsdl'));
        $object = $client->__soapCall('purgeObject', array(
            $params
        ));
        unset($form_state['storage']['confirm']);
    }
    catch (exception $e)
    {
        if (preg_match('/org\.fcrepo\.server\.security\.xacml\.pep\.AuthzDeniedException/', $e->getMessage()))
        {
            drupal_set_message(t('Error: Insufficient permissions to purge object.'), 'error');
        }
        else
        {
            drupal_set_message(t($e->getMessage()), 'error');
        }
        return;
    }
}
if (!empty($form_state['values']['referrer']))
{
    $form_state['redirect'] = $form_state['values']['referrer'];
}
elseif (empty($collectionPid) && !empty($_SESSION['fedora_collection']) && $_SESSION['fedora_collection'] !=
$pid)
{
    $collectionPid = $_SESSION['fedora_collection'];
    $form_state['redirect'] = "fedora/repository/$collectionPid/";
}
else
{
    $form_state['redirect'] = 'fedora/repository/';
}
}
}
function islandora_book_help($path, $arg)
{
    switch ($path)
    {
        case 'admin/help#islandora_book':
            return t('<p>
The Islandora Book Module allows users to create digital representations of books within the Fedora
Repository.
A book object is created using !mods data. A collection of associated page objects may then be created
created. Page objects
are created by ingesting a zipped folder of TIFF images, each representing a distinct page.

```

```

    Page TIFFs are used to create a series of Fedora datastreams on ingestion.  These streams include
    derivative images
    for display - JPEGs, Thumbnails, and JP2s, and one ore more OCR streams for text display and analysis.
    </p>
    <p>
    <h3>Configuration</h3>
    The page datastreams may be created on this server, or on an external processing server with !
    microservices.<br/>
    To process images from TIFFS on this server check <strong>Create derivative images locally?</strong> on
    the admin page. (Link at the bottom of this page).<br />
    TO create OCR streams on this server check <strong>Perform OCR on incoming TIFF images?</strong>. <br
    /><br />

    , array(
    !microservices' => 1("Fedora Microservices", 'https://github.com/Islandora/fedora_microservices'),
    !mods' => 1('MODS', 'http://www.loc.gov/standards/mods/')
    ));
}
}

```

6.1.10 sites/all/modules/islandora-github/xsl/sparql_to_html.xsl

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:s="http://www.w3.org/2001/sw/DataAccess/rf1/result" xmlns:php="http://php.net/xsl" version="1.0"
  exclude-result-prefixes="php">
  <!-- Red and White XSLT -->
  <xsl:variable name="BASEURL" select="$baseUrl"/>
  <xsl:variable name="PATH" select="$path"/>
  <xsl:variable name="thisPid" select="$collectionPid"/>
  <xsl:variable name="size" select="20"/>
  <xsl:variable name="page" select="$hitPage"/>
  <xsl:variable name="start" select="(number($page) - 1) * number($size) + 1"/>
  <xsl:variable name="end" select="($start - 1) + number($size)/>
  <xsl:variable name="cellsPerRow" select="4"/>
  <xsl:variable name="count" select="count(s:sparql/s:results/s:result)"/>
  <xsl:variable name="totpage" select="floor(($count - 1) div $size) + 1"/>
  <xsl:template match="/">
    <xsl:if test="$count>0">
      <xsl:call-template name="render_pager"/>
      <!-- <table class="collview" cellpadding="3" cellspacing="3" width="90%" -->
      <table class="collview">
        <xsl:apply-templates select="s:sparql/s:results"/>
      </table>
      <br clear="all"/>
      <xsl:call-template name="render_pager"/>
    </xsl:if>
  </xsl:template>
  <xsl:template match="s:sparql/s:results">
    <xsl:for-each select="s:result[position() mod $cellsPerRow = 1 and position()&gt;=$start and position()
    &lt;=$end]">
      <tr>
        <xsl:apply-templates select=". | following-sibling::s:result[position() &lt; $cellsPerRow]" />
      </tr>
    </xsl:for-each>
  </xsl:template>
  <xsl:template name="pageNumbers">
    <xsl:param name="pageIndex"/>
    <xsl:variable name="distanceFromCurrent" select="$pageIndex - $page"/>
    <xsl:variable name="endIndex" select="$pageIndex * $size"/>
    <xsl:variable name="startIndex" select="$endIndex - $size + 1"/>
    <!-- show a maximum of nine paged sets on either side of the current paged set; just like Google does it -->
    <xsl:choose>
      <xsl:when test="($distanceFromCurrent &gt; -5 and $distanceFromCurrent &lt; 5)">
        <xsl:choose>
          <xsl:when test="$pageIndex = $page">
            <li class="pager-current">
              <xsl:choose>
                <xsl:when test="$pageIndex = $totpage">
                  <xsl:value-of select="concat($startIndex, '-', $count)"/>
                </xsl:when>
                <xsl:otherwise>
                  <xsl:value-of select="concat($startIndex, '-', $endIndex)"/>
                </xsl:otherwise>
              </xsl:choose>
            </li>
          </xsl:when>
          <xsl:otherwise>
            <li class="pager-item">
              <a>
                <xsl:attribute name="href">
                  <xsl:value-of select="concat($BASEURL, '/fedora/repository/', $thisPid, '/-/Collection/',
                  $pageIndex)"/>
                </xsl:attribute>
                <xsl:choose>
                  <xsl:when test="$pageIndex = $totpage">
                    <xsl:value-of select="concat($startIndex, '-', $count)"/>
                  </xsl:when>
                  <xsl:otherwise>
                    <xsl:value-of select="concat($startIndex, '-', $endIndex)"/>
                  </xsl:otherwise>
                </xsl:choose>
              </a>
            </li>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:choose>
    </xsl:template>

```

```

</xsl:choose>
</xsl:when>
<xsl:when test="($distanceFromCurrent = -5)">
  <li class="pager-previous"><a><xsl:attribute name="href"><xsl:value-of select="concat($BASEURL,
'/fedora/repository/', $thisPid, '/-/Collection/', 1)"/></xsl:attribute>
  1 -...
  </a>
</li>
</xsl:when>
<xsl:when test="($distanceFromCurrent = 5)">
  <li class="pager-next">
    <a>
      <xsl:attribute name="href">
        <xsl:value-of select="concat($BASEURL, '/fedora/repository/', $thisPid, '/-/Collection/',
$totpage)"/>
      </xsl:attribute>
      <xsl:value-of select="concat('...- ', $count)"/>
    </a>
  </li>
</xsl:when>
</xsl:choose>
<!-- recursively call the template for all the paged sets -->
<xsl:if test="$pageIndex * $size < $count">
  <xsl:call-template name="pageNumbers">
    <xsl:with-param name="pageIndex" select="$pageIndex + 1"/>
  </xsl:call-template>
</xsl:if>
</xsl:template>
<xsl:template name="render_pager">
  <!-- start previous next -->
  <div class="item-list">
    <ul class="pager">
      <xsl:choose>
        <xsl:when test="$end &gt;= $count and $start = 1">
          <xsl:call-template name="pageNumbers">
            <xsl:with-param name="pageIndex" select="1"/>
          </xsl:call-template>
        </xsl:when>
        <xsl:when test="$end &gt;= $count">
          <xsl:call-template name="pageNumbers">
            <xsl:with-param name="pageIndex" select="1"/>
          </xsl:call-template>
        </xsl:when>
        <xsl:when test="$start = 1">
          <xsl:call-template name="pageNumbers">
            <xsl:with-param name="pageIndex" select="1"/>
          </xsl:call-template>
        </xsl:when>
        <xsl:otherwise>
          <xsl:call-template name="pageNumbers">
            <xsl:with-param name="pageIndex" select="1"/>
          </xsl:call-template>
        </xsl:otherwise>
      </xsl:choose>
    </ul>
  </div>
  <!-- end previous next-->
</xsl:template>
<xsl:template match="s:result">
  <xsl:variable name="OBJECTURI" select="s:object/@uri"/>
  <xsl:variable name="CONTENTURI" select="s:content/@uri"/>
  <xsl:variable name="CONTENTMODEL" select="substring-after($CONTENTURI, '/')"/>
  <xsl:variable name="PID" select="substring-after($OBJECTURI, '/')"/>
  <xsl:variable name="newTitle">
    <xsl:call-template name="replace-string">
      <xsl:with-param name="text" select="s:title"/>
      <xsl:with-param name="from" select="' '"/>
      <xsl:with-param name="to" select="'&#x201C;"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:variable name="cleanTitle" select="php:functionString('fedora_repository_urlencode_string',
$newTitle)"/>
  <xsl:variable name="linkUrl">
    <xsl:choose>
      <xsl:when test="($CONTENTMODEL='islandora:collectionCMModel')">
        <xsl:value-of select="concat($BASEURL, '/fedora/repository/', $PID, '/-/collection')"/>
      </xsl:when>
      <xsl:otherwise>
        <!--the below is an example of going straight to a datastream instead of the details page.
<xsl:value-of select="$BASEURL"/>/fedora/repository/<xsl:copy-of select="$PID"/>/OBJ/<xsl:value-of
select="s:title"/>-->
        <xsl:value-of select="concat($BASEURL, '/fedora/repository/', $PID)"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:value-of select="s:content"/>
</xsl:variable>
<xsl:choose>
  <xsl:when test="($CONTENTMODEL='islandora:collectionCMModel')">
    <td valign="top" width="25%">
      <table class="colcell1">
        <tr>
          <td class="colldescri">
            <xsl:value-of select="$newTitle" disable-output-escaping="yes"/>
          </td>
        </tr>
      </table>
    </td>
  </xsl:when>
</xsl:choose>

```

```

        <td class="cellthumb1">
            <a>
                <xsl:attribute name="href">
                    <xsl:value-of select="$linkUrl"/>
                </xsl:attribute>
                <img>
                    <xsl:attribute name="src">
                        <xsl:value-of select="concat($BASEURL, '/fedora/repository/', $PID, '/TN')"/>
                    </xsl:attribute>
                    <xsl:attribute name="alt">
                        <xsl:value-of select="$newTitle" disable-output-escaping="yes"/>
                    </xsl:attribute>
                </img>
            </a>
        </xsl:if>
    </td>
</tr>
</table>
</td>
</xsl:when>
<xsl:otherwise>
    <td valign="top" width="25%">
        <table class="collcell">
            <tr>
                <td class="cellthumb" height="250px">
                    <a>
                        <xsl:attribute name="href">
                            <xsl:value-of select="$linkUrl"/>
                        </xsl:attribute>
                        <img>
                            <xsl:attribute name="src">
                                <xsl:value-of select="concat($BASEURL, '/fedora/repository/', $PID, '/TN')"/>
                            </xsl:attribute>
                            <xsl:attribute name="alt">
                                <xsl:value-of select="$newTitle" disable-output-escaping="yes"/>
                            </xsl:attribute>
                        </img>
                    </a>
                    <br clear="all"/>
                </td>
            </tr>
            <tr>
                <td class="celldescr">
                    <xsl:choose>
                        <xsl:when test="($CONTENTMODEL='islandora:pageCModel')">
                            page <xsl:value-of select="substring-after($newTitle, '- page')>
                            <xsl:attribute name="page" select="substring-after($newTitle, '- page')>
                            </xsl:when>
                        <xsl:otherwise>
                            <xsl:value-of select="$newTitle" disable-output-escaping="yes"/>
                        </xsl:otherwise>
                    </xsl:choose>
                </td>
            </tr>
        </table>
    </td>
</xsl:otherwise>
</xsl:choose>
<xsl:if test="(position() = last()) and (position() < $cellsPerRow)">
    <xsl:call-template name="FillerCells">
        <xsl:with-param name="cellCount" select="$cellsPerRow - position()"/>
    </xsl:call-template>
</xsl:if>
</xsl:template>
<xsl:template name="FillerCells">
    <xsl:param name="cellCount"/>
    <td> </td>
    <xsl:if test="$cellCount > 1">
        <xsl:call-template name="FillerCells">
            <xsl:with-param name="cellCount" select="$cellCount - 1"/>
        </xsl:call-template>
    </xsl:if>
</xsl:template>
<xsl:template name="replace-string">
    <xsl:param name="text"/>
    <xsl:param name="from"/>
    <xsl:param name="to"/>
    <xsl:choose>
        <xsl:when test="contains($text, $from)">
            <xsl:variable name="before" select="substring-before($text, $from)"/>
            <xsl:variable name="after" select="substring-after($text, $from)"/>
            <xsl:variable name="prefix" select="concat($before, $to)"/>
            <xsl:value-of select="$before"/>
            <xsl:value-of select="$to"/>
            <xsl:call-template name="replace-string">
                <xsl:with-param name="text" select="$after"/>
                <xsl:with-param name="from" select="$from"/>
                <xsl:with-param name="to" select="$to"/>
            </xsl:call-template>
        </xsl:when>
        <xsl:otherwise>
            <xsl:value-of select="$text"/>
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>
</xsl:stylesheet>

```


6.1.11 sites/all/modules/islandora_solr_search/IslandoraSolrResults.inc

```

<?php
/**
 * @file
 * Contains methods to search solr and display results. depends on Apache_Solr_Php client.
 */
/**
 * Islandora Solr Results
 */
class IslandoraSolrResults
{
    static $facetSeparator = '~'; //used to separate facets in url
    static $slashReplacement = '~slash~'; // a pattern used to replace / in url's the slash breaks drupal clean url's
    public static $SEARCH_CLASS_ADVANCED_SEARCH_NUMBER_FIELDS = 5;
    public $facetFieldArray = array();
    public $searchFieldArray = array();
    public $resultFieldArray = array();
    public $allSubsArray = array();
    /**
     * Constructor
     */
    function IslandoraSolrResults ()
    {
        module_load_include('php', 'islandora_solr_search', 'SolrPhpClient/Apache/Solr/Service');
        module_load_include('inc', 'islandora_solr_search', 'includes/common');
        $this->prepFieldSubstitutions();
    }
    /**
     * Output the main body of the search results
     *
     * @param type $solrQueryProcessor
     * @param type $title
     * @param type $output
     * @return string
     */
    function displayResults($solrQueryProcessor, $title = "Search Results", $output = '')
    {
        $apacheSolrResult = $solrQueryProcessor->solrResult;
        $total = (int) $apacheSolrResult->response->numFound;
        $end = min(($solrQueryProcessor->solrLimit + $solrQueryProcessor->solrStart), $total);
        // Check for empty resultset
        if ($total === 0)
        {
            $output = "<h2>Search Results</h2>";
            $output .= "<div>Sorry, but your search returned no results.</div>";
            return $output;
        }
        // Initialize drupal-style pager
        islandora_solr_search_pager_init($total, $solrQueryProcessor->solrLimit);
        // Get secondary display profiles
        $secondary_display_profiles = module_invoke_all('islandora_solr_secondary_display');
        // How about this for an expected return?
        // return array(
        // 'machine-name' = array(
        // 'name' => 'Human Readable Name',
        // 'module' => 'module_name',
        // 'file' => 'FileName.inc',
        // 'class' => 'ClassName',
        // 'function' => 'function_name',
        // 'description' => 'A description of the display profile',
        // 'logo' => 'Some text, some <html>, an <img>, etc used to link to this output,
        // );
        // check if
        // if the islandora_solr_search admin form has never been saved, then
        // the variable 'islandora_solr_secondary_display' won't be set yet.
        $secondary_array = variable_get('islandora_solr_secondary_display', array());
        // count number of elements in array
        $array_count = count($secondary_array);
        // count number of disabled displays
        $i = 0;
        foreach ($secondary_array as $key => $value)
        {
            if ($value === 0)
            {
                $i++;
            }
        }
        // if number of elements is the same as the number of disabled displays,
        // this means that there are no displays enabled.
        if ($array_count != $i)
        {
            $output .= "<div id='islandora_solr_secondary_display_profiles'>";
            foreach ($secondary_array as $name => $status)
            {
                if ($status === $name)
                {
                    $description = $secondary_display_profiles[$name]['description'];
                    $logo = $secondary_display_profiles[$name]['logo'];
                    $attributes = array(
                        'title' => $description
                    );
                    $query = array();
                    $query[] = drupal_query_string_encode(array(
                        'solr_profile' => $name
                    ));
                }
            }
        }
    }
}

```

```

    ));
    $output .= l($logo, $_GET['q'], array(
        'attributes' => $attributes,
        'html' => true,
        'query' => count($query) ? implode('&', $query) : NULL
    ));
}
$output .= "</div>";
}
$output .= '<h2>' . $title . '</h2>';
$output .= '<div id="islandora_solr_result_count">(' . ($solrQueryProcessor->solrStart + 1) . ' - ' . $end .
' of ' . $total . ')</div>';
$output .= '<div class="content">';
// drupal_set_message(print_r($secondary_display_profiles,1));
/*
 * Then, here, we need to build links to the current query under the secondary query display
 * profile for each such profile which uses the hook.
 *
 * The link should basically be $theCurrentURL with $proile=<profile_name> tacked on.
 *
 * Also, perhaps there's no difference between primary and secondary profiles?
 */
$output .= $this->printResults($solrQueryProcessor->solrResult);
// Render Drupal pager (bottom of search results)
$output .= theme('pager', null, $solrQueryProcessor->solrLimit, 0, null, 5);
$output .= "</div>"; // class = "content"
return $output;
}
/**
 * Function: printResults
 *
 * Description: translates a solr query result into a basic in-browser search result.
 *
 * @param Apache_Solr_Object $results
 */
function printResults($results)
{
    $items = array();
    $type = "ol";
    $title = null;
    $recordStart = $results->response->start;
    $limitResults = variable_get('islandora_solr_search_limit_result_fields', 0);
    global $base_url;
    if (empty($results))
    {
        return "no results";
    }
    foreach ($results->response->docs as $doc)
    {
        $rows = array();
        $row = 0;
        foreach ($doc as $field => $value)
        {
            if ($limitResults && empty($this->resultFieldArray[$field]))
            {
                continue;
            }
            $translated_field_name = isset($this->allSubsArray[$field]) ? $this->allSubsArray[$field] : $field;
            $rows[$row][] = array(
                'data' => $translated_field_name,
                'header' => true
            );
            if (is_array($value))
            {
                $value = implode(", ", $value);
            }
            if ($field == 'PID')
            {
                $l = l($value, 'fedora/repository/' . htmlspecialchars($value, ENT_QUOTES, 'utf-8'));
                $rows[$row][] = $l;
            }
            else
            {
                $rows[$row][] = $value;
            }
            $row++;
        }
        $items[] = theme_table(null, $rows, array(
            'class' => 'islandora_solr_search_results_object'
        ));
    }
    $output .= theme_item_list($items, $title, $type, array(
        'class' => 'stuff',
        'start' => ($recordStart + 1)
    ));
}
if (variable_get('islandora_solr_search_debug_mode', 0)) // debug dump
{
    $results_r .= "<pre>Results: " . print_r($results, true) . "</pre>";
    $fieldset_r = array(
        '#title' => t("Raw Results"),
        '#collapsible' => TRUE,
        '#collapsed' => TRUE,
        '#value' => $results_r
    );
    $output .= theme('fieldset', $fieldset_r);
}

```

```

    }
    return $output;
}
/**
 * Function: currentQuery
 *
 * Description: Displays elements of the current solr query (with enabled filters).
 * This functionality was previously rolled into the facet block, but it makes
 * sense to separate the two functions, particularly as the current query is now
 * imprinted in the system's breadcrumb.
 *
 * @param IslandoraSolrQueryProcessor $solrQueryProcessor
 */
function currentQuery($solrQueryProcessor)
{
    //dsm($solrQueryProcessor);
    global $base_url;
    $output = '';
    $islandora_fq = restoreSlashes($solrQueryProcessor->solrFilters);
    $islandora_facets = $solrQueryProcessor->solrResult->facet_counts->facet_fields;
    //dsm($islandora_facets);
    if (strlen(trim($solrQueryProcessor->solrQuery))
    {
        // Variables to build a drupal-ish unordered list
        $query_list_title = "Query";
        $query_list_items = array();
        $query_list_type = "ul";
        $query_list_attributes = array(
            'class' => 'islandora_solr_search_query_list query_list'
        );
        // OK, so it's a list of one.
        // In the future, we could reasonably break the query on boolean operators
        // and allow one part to be removed at a time.
        $query_list_items[] .= 1('',
'islandora/solr/search/' . " " . '/' . replaceSlashes($islandora_fq) . '/dismax', array(
    'html' => true
    )) . ' ' . stripslashes($solrQueryProcessor->solrQuery);
    $output .= '<div class="islandora_solr_search_query_wrap">';
    $output .= theme_item_list($query_list_items, $query_list_title, $query_list_type, $query_list_attributes);
    $output .= '</div>'; // 'islandora_solr_search_query_wrap'
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Here we will build links to remove currently enforced facet filters ... //
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
$filters = empty($islandora_fq) ? null : explode(IslandoraSolrResults::$facetSeparator, $islandora_fq);
if (!empty($filters) && $islandora_fq != '-')
{
    // $filter_output = null;
    // Variables to build a drupal-ish unordered list, which will be rendered
    // a little bit farther down
    $filter_list_title = "Enabled Filters";
    $filter_list_items = array();
    $filter_list_type = "ul";
    $filter_list_attributes = array(
        'class' => 'islandora_solr_search_filter_list filter_list'
    );
    foreach ($filters as $key => $filter)
    {
        $edit_filters = $filters;
        unset($edit_filters[$key]);
        $filter_disable = implode(IslandoraSolrResults::$facetSeparator, $edit_filters);
        if (empty($filter_disable))
        {
            $filter_disable = '-';
        }
        $filter_name_and_value = explode(":", $filter, 2);
        $filter_name = $filter_name_and_value[0];
        $filter_value = $filter_name_and_value[1];
        if (strcmp($filter_name, '-') < 1)
            $exclude = false;
        else
        {
            $exclude = true;
            $filter_name = substr($filter_name, 1);
        }
        $filter_list_items[] = 1('',
'islandora/solr/search/' . replaceSlashes($solrQueryProcessor->solrQuery) . '/' .
replaceSlashes($filter_disable) . '/' . $solrQueryProcessor->solrDefType, array(
    'html' => true
    )) . ' ' . $this->facetFieldArray[$filter_name] . ($exclude ? ' != ' : ' = ') . $filter_value;
    }
}
if (!empty($filter_list_items))
{
    // $filter_output = str_replace('%2B','%252B',$filter_output);
    $output .= '<div class="islandora_solr_search_filter_wrap">';
    $output .= theme_item_list($filter_list_items, $filter_list_title, $filter_list_type,
$filter_list_attributes);
    $output .= '</div>'; // 'islandora_solr_search_filter_wrap'
}
return $output;
}
/**
 * Function: prepFieldSubstitutions

```

```

*
* Description: reads configuration values and preps a number of key=>value
* arrays for output substitution
*/
function prepFieldSubstitutions()
{
  $rawFacetVals = variable_get("islandora_solr_search_block_facets", 'dc.subject ~ Subject,dc.type ~ Type');
  $this->facetFieldArray = islandora_build_substitution_list($rawFacetVals);
  $rawSearchTerms = variable_get('islandora_solr_searchterms', 'dc.title ~ Title');
  $this->searchFieldArray = islandora_build_substitution_list($rawSearchTerms);
  $rawResultFields = variable_get('islandora_solr_search_result_fields', 'dc.subject ~ Subject,dc.type ~
Type');
  $this->resultFieldArray = islandora_build_substitution_list($rawResultFields);
  $this->allSubsArray = array_merge($this->facetFieldArray, $this->searchFieldArray, $this->resultFieldArray);
}
/**
* Function: displayFacets
*
* Description: Displays basic facets based on an apache solr query response,
* as contained with the IslandoraSolrQueryProcessor.
*
* @global string $base_url
* @param IslandoraSolrQueryProcessor $solrQueryProcessor
* @return string
*/
function displayFacets($solrQueryProcessor)
{
  global $base_url;
  $output = '';
  $islandora_fq = $solrQueryProcessor->solrFilters;
  $islandora_facets = $solrQueryProcessor->solrResult->facet_counts->facet_fields;
  if (empty($islandora_facets))
  {
    return $output; //no facets to show
  }
  foreach ($islandora_facets as $key => $field)
  {
    $list_items = array();
    $list_type = "ul";
    $list_title = null;
    $test = get_object_vars($field); //get the number of fields if there aren't any don't show the key
    if (count($test) > 0)
    {
      $facet_count = 0;
      unset($normal_facet_output);
      $filter_include = null;
      $filter_exclude = null;
      foreach ($field as $name => $number)
      {
        if ($islandora_fq && $islandora_fq != '-') //there are existing facets in the query
        {
          $disable_link = strstr($islandora_fq, $key . ':' . replaceSlashes($name) . ''); //we don't want a
link for this facet as we already used it
          if ($disable_link)
          {
            //don't show link to this facet but include a link to remove it
          }
          else
          {
            $filter_include = $key . ':' . '' . $name . '' . IslandoraSolrResults::$facetSeparator .
$islandora_fq;
            $filter_exclude = '-' . $key . ':' . '' . $name . '' . IslandoraSolrResults::$facetSeparator .
$islandora_fq;
          }
        }
        else //no existing facets chosen
        {
          $filter_include = $key . ':' . '' . $name . '';
          $filter_exclude = '-' . $key . ':' . '' . $name . '';
        }
        $filter_include = replaceSlashes($filter_include); //replace the slash so url does not break
        $filter_exclude = replaceSlashes($filter_exclude); //replace the slash so url does not break
        if ($disable_link)
        {
          // we don't want to create a link, because we're already filtering on this value.
          // Links to remove enabled facet filters are created down below.
        }
        else //normal link
        {
          $evenodd = ($facet_count % 2 ? 'odd' : 'even');
          $lplus = 'islandora/solr/search/' . $solrQueryProcessor->solrQuery . '/' . $filter_include . '/' .
$solrQueryProcessor->solrDefType;
          $lminus = 'islandora/solr/search/' . $solrQueryProcessor->solrQuery . '/' . $filter_exclude . '/' .
$solrQueryProcessor->solrDefType;
          $text = $name;
          $attrs = array();
          $list_items[] = l($text, $lplus, $attrs) . ' (' . $number . ')'. '<span class="plusminus">' . l('+',
$lplus, array(
            'attributes' => array(
              'class' => 'plus'
            )
          )) . ' ' . l('-', $lminus, array(
            'attributes' => array(
              'class' => 'minus'
            )
          )) . '</span>';
        }
      }
    }
  }
}

```



```

        '#title' => '',
        '#type' => 'select',
        '#default_value' => $andorj,
        '#options' => $andOrArray
    );
    /*
    $form['search_type'][$type$i] = array(
        '#title' => '',
        '#type' => 'select',
        '#options' => $types,
        '#default_value' => trim($field_and_term[0])
    );
    $form["fedora_terms$i"] = array(
        '#size' => '24',
        '#type' => 'textfield',
        '#title' => '',
        '#default_value' => (count($field_and_term) >= 2 ? trim($field_and_term[1], '()') : ''),
    );
    */
    }
}
$form['submit'] = array(
    '#type' => 'submit',
    '#value' => t('Search')
);
return $form;
}
/**
 * Build simple solr form
 * @global type $islandora_query
 * @return type
 */
function build_simple_solr_form()
{
    $required = FALSE;
    if (variable_get('islandora_solr_search_block_request_handler', '') == 'standard')
    {
        $required = TRUE;
    }
    global $islandora_query;
    if (!strpos($islandora_query, ':'))
    {
        $default = $islandora_query;
    }
    // TODO: Fix the above. The global is no longer valid, and the test is not conclusive. Should instead test
    for deftype=dismax.
    $form["islandora_simple_search_query"] = array(
        '#size' => '24%',
        '#type' => 'textfield',
        '#title' => '',
        '#default_value' => $default,
        '#required' => $required
    );
    $form['submit'] = array(
        '#type' => 'submit',
        '#value' => t('search')
    );
    return $form;
}
/**
 * Theme solr search form
 * @param type $form
 * @return type
 */
function theme_solr_search_form($form)
{
    if (!isset($repeat))
    {
        $repeat = variable_get('islandora_solr_search_block_repeat', '3');
    }
    $output = drupal_render($form['search_type']['type1']);
    $output .= drupal_render($form['fedora_terms1']);
    // $output .= drupal_render($form['andor1']) . drupal_render($form['search_type']['type2']);
    $output .= drupal_render($form['andor1']);
    /*
    $output .= drupal_render($form['fedora_terms2']);
    if ($repeat > 2 && $repeat < 9) {
        for ($i = 3; $i < $repeat + 1; $i++) {
            $t = $i - 1;
            $output .= drupal_render($form["andor$t"]) . drupal_render($form['search_type'][$type$i]);
            $output .= drupal_render($form["fedora_terms$i"]);
        }
    }

    $output .= drupal_render($form['submit']);
    */
    $output .= drupal_render($form);
    return $output;
}
}

```

6.1.12 sites/all/modules/islandora_solr_search/IslandoraSolrQueryProcessor.inc

```

<?php
/**
 * @file
 * Contains methods to build and execute a solr query.
 * Depends on Apache_Solr_Php client.
 */
/**
 * Islandora Solr Query Processor
 * @todo Stop using global vars, start using this object's vars.
 * The module itself should initially create an instance of this
 * object and reference the query vars inside that object when calling
 * other output display functions.
 */
class IslandoraSolrQueryProcessor
{
    static $facetSeparator = '~'; //used to separate facets in url
    static $slashReplacement = '~slsh~'; // a pattern used to replace / in url's the slash breaks drupal clean url's
    public static $SEARCH_CLASS_ADVANCED_SEARCH_NUMBER_FIELDS = 5;
    public $solrQuery;
    public $solrParams;
    public $solrStart;
    public $solrLimit;
    public $solrDefType;
    public $solrFilters;
    public $solrResult;
    /**
     * Constructor
     */
    function IslandoraSolrQueryProcessor()
    {
        module_load_include('php', 'islandora_solr_search', 'SolrPhpClient/Apache/Solr/Service');
        module_load_include('inc', 'islandora_solr_search', 'includes/common');
    }
    /**
     * Build and execute a query
     * @param type $query
     * @param type $fq
     * @param type $dismax
     * @return type
     */
    function buildAndExecuteQuery($query, $fq = null, $dismax = null)
    {
        //
        if (variable_get('islandora_solr_search_block_request_handler', NULL) == 'standard')
        {
            $namespace = trim(variable_get('solr_namespace_restriction', NULL));
            if (!$query || $query == '')
            {
                $query = '%252F';
            }
            else
            {
                if ((strpos($query, 'OCR.OCR')) === FALSE)
                {
                    if (strpos($fq, "rels.hasModel:bookCModel") === FALSE)
                    {
                        if ($fq != NULL && $fq != '-')
                        {
                            $fq .= $this->facetSeparator . "rels.hasModel:bookCModel";
                        }
                        else
                        {
                            $fq = "rels.hasModel:bookCModel";
                        }
                    }
                }
            }
        }
        $this->buildQuery($query, $fq, $dismax);
        return $this->executeQuery();
    }
    /**
     * Default implementation for solr search. You can configure others in the block config settings
     * @todo Get rid of the global $facetArray? It is the only global yet relied upon.
     */
    function buildQuery($query, $fq = null, $dismax = null)
    {
        // Get pager variable
        $startPage = isset($_GET['page']) ? $_GET['page'] : 0;
        $this->solrLimit = variable_get('islandora_solr_search_num_of_results', 20);
        global $base_url, $facetArray; //used by facet block as well
        // -- cut here --
        // Get display profile
        $islandora_solr_primary_display = variable_get('islandora_solr_primary_display', 'default');
        $profiles = module_invoke_all("islandora_solr_primary_display");
        if (isset($profiles[$islandora_solr_primary_display]))
        {
            drupal_set_message("There is an error in the solr search configuration: the display profile is not found.", 'error');
            return "";
        }
        $profile = $profiles[$islandora_solr_primary_display];
        // Include the file for the display profile
    }
}

```

```

require_once(drupal_get_path('module', $profile['module']) . '/' . $profile['file']);
// Set display class and function vars
$solrClass = $profile['class'];
$solrFunction = $profile['function'];
// -- end cut here --
//fix the query as some characters will break the search : and / slash are examples
$this->solrQuery = restoreSlashes($query);
if (empty($this->solrQuery))
{
  // $this->solrQuery = '%20'; //so we can allow empty queries to dismax
}
$facetArray = array();
$facetFields = '';
$rawFacetVals = variable_get("islandora_solr_search_block_facets", 'dc.subject,dc.type');
$facetArray = islandora_build_substitution_list($rawFacetVals);
$facetFields = implode(",", array_keys($facetArray));
$facetLimit = variable_get('islandora_solr_search_block_facet_limit', '12');
$facetMinCount = variable_get('islandora_solr_search_block_facet_min_count', '2');
$requestHandler = variable_get("islandora_solr_search_block_request_handler", "standard");
$this->solrParams = array(
  'facet' => 'true',
  'facet.mincount' => $facetMinCount,
  'facet.limit' => $facetLimit,
  'qt' => $requestHandler,
  'facet.field' => explode(',', $facetFields) //comma separated list configured in the block config
);
$debugQuery = (variable_get('islandora_solr_search_debug_mode', 0) ? "true" : null); //any val incl. 'false'
is treated as true by Solr
// $fq = preg_replace('/PID:.*\*/', '', $fq);
if ($fq != NULL && $fq != '-')
{
  $fq = restoreSlashes($fq); //put the slash back
  $fq = csv_explode(IslandoraSolrQueryProcessor::$facetSeparator, $fq, '"', true); //to filter by more than
  one facet we will separate them by~ for nowseparate them by~ for now
  $this->solrParams['fq'] = $fq;
  $islandora_fq = replaceSlashes($fq); //remove the slash here as we will be using this in url's
}
if (empty($islandora_fq))
{
  $islandora_fq = '-';
}
if ($dismax != NULL)
{
  $this->solrDefType = $dismax;
  $this->solrParams['defType'] = $dismax;
}
$this->solrStart = max(0, $startPage) * $this->solrLimit;
// The breadcrumb should go in the display class
$queryurl = "islandora/solr/search/" . replaceSlashes($this->solrQuery);
$breadcrumb_fq = $islandora_fq;
if (strcmp($islandora_fq, "-"))
{
  foreach (csv_explode(IslandoraSolrQueryProcessor::$facetSeparator, $islandora_fq, '"', true) as $facet)
  {
    $field = substr($facet, 0, strpos($facet, ":"));
    if ($field != "rels.hasModel")
    {
      $value = restoreSlashes(trim(substr($facet, strpos($facet, ":") + 1, ''));
      $options = array(
        'html' => true
      );
      $options['attributes']['title'] = $facet;
      $crumbLink = $queryurl . "/" . $breadcrumb_fq . (empty($this->solrDefType) ? '' : '/' . $this->solrDefType);
      $cut_fq = $this->delete_filter($islandora_fq, $facet);
      $cutLink = $queryurl . "/" . $cut_fq . (empty($this->solrDefType) ? '' : '/' . $this->solrDefType);
      if (!strcmp("-", $facet, 1))
      {
        $options['attributes']['class'] = "strikethrough";
      }
      $breadcrumb[] = l($value, $crumbLink, $options) . " <span
class='islandora_solr_search_breadcrumb_super'>(" . l("x", $cutLink, array(
        'attributes' => array(
          'title' => "Remove " . $facet
        )
      )) . "</span>";
      $breadcrumb_fq = $this->delete_filter($breadcrumb_fq, $facet);
    }
  }
}
if (!empty($this->solrQuery) && strcmp($this->solrQuery, ' '))
{
  $cutLink = "islandora/solr/search/" . $islandora_fq . "/dismax";
  $queryval = $this->solrQuery;
  $tokens = explode(" ", $queryval);
  foreach ($tokens as $token)
  {
    preg_match("/^.*:/", $token, $matches);
    $queryval = str_replace($matches[0], "", $queryval);
  }
  if ($queryval != '%252F')
  {
    $elencoparole = preg_replace('~\s+~', ' ', trim($queryval));
    $searchwords = explode(" ", $elencoparole);
    $elencoparole = "<strong>" . $searchwords[0] . "</strong>";
    $nparole = count($searchwords);
  }
}

```

```

        if ($nparole > 2)
        {
            for ($i = 1; $i < ($nparole - 1); $i++)
            {
                $elencoparole .= " " . $searchwords[$nparole - 1] . " " . "<strong>" . $searchwords[$i] .
            }
        }
        $elencoparole .= " <span class='islandora_solr_search_breadcrumb_super'>°</span>";
        $breadcrumb[] = $elencoparole;
        // $breadcrumb[] = l($queryval, $queryurl . "/" . $this->solrQuery);
        // $breadcrumb[] = l($queryval, $queryurl . "/" . $this->solrQuery);
        // <span class='islandora_solr_search_breadcrumb_super'>(" . l("x", $cutlink,
        array('attributes' => array('title' => "Remove " . $facet)) . ")</span>";
    }
}
// $breadcrumb[] = l('Home', NULL);
$breadcrumb[] = "Search";
if (!empty($breadcrumb))
    $breadcrumb = array_reverse($breadcrumb);
drupal_set_breadcrumb($breadcrumb);
$this->solrFilters = $islandora_fq;
// if (strpos($fq, "rels.hasModel%253AbookCModel") === FALSE) {
//     if ($fq) {
//         $fq .= $this->facetSeparator;
//     }
//     $fq = "rels.hasModel%253AbookCModel";
// }
// At this point let's invoke a hook for third-party modules to mess about
// with the query parameters if they need to. Third party modules may alter
// this object's query parameters if they wish.
// module_invoke_all("islandora_solr_search_query_processor", &$this);
return;
}
/**
 * Reset Results
 */
function resetResults()
{
    unset($this->solrResult);
}
/**
 * Execute the query
 * @return type
 */
function executeQuery()
{
    // ////////////////////////////////////////
    // Right here the function should be split. One function to execute the query, //
    // another to set up pager variables and generate display. //
    // ////////////////////////////////////////
    $url = variable_get('islandora_solr_search_block_url', 'http://localhost:8080/solr');
    $pathParts = parse_url($url);
    $solr = new Apache_Solr_Service($pathParts['host'], $pathParts['port'], $pathParts['path'] . '/');
    $solr->setCreateDocuments(0);
    $this->solrParams['hl'] = 'true';
    $this->solrParams['hl.fl'] = '*';
    $this->solrParams['hl.requireFieldMatch'] = 'true';
    $this->solrParams['hl.snippets'] = '10';
    $this->solrParams['hl.simple.pre'] = '<b>';
    $this->solrParams['hl.simple.post'] = '</b>';
    $bcksolrQuery = $this->solrQuery;
    // if (strpos($bcksolrQuery, 'OCR.OCR') !== FALSE) {
    $bcksolrQuery = preg_replace('~\s+~', ' ', trim($bcksolrQuery));
    $termini = explode(" ", $bcksolrQuery);
    $parole = explode(" ", trim($termini[1]));
    $numeroparole = count($parole);
    $bcksolrQuery = "";
    // $bcksolrQuery = "(OCR.OCR:" . $parole[0] . ")";
    $bcksolrQuery = "(" . $termini[0] . ":" . $parole[0] . ")";
    if ($numeroparole > 1)
    {
        for ($i = 1; $i < $numeroparole; $i++)
        {
            // $bcksolrQuery = $bcksolrQuery . $termini[2] . "(OCR.OCR:" . $parole[$i] .
        }
        $bcksolrQuery = $bcksolrQuery . $termini[2] . "(" . $termini[0] . ":" . $parole[$i] . ")";
    }
    // }
    // This is where the query gets executed and output starts being created.
    try
    {
        $results = $solr->search($this->solrQuery, $this->solrStart, $this->solrLimit, $this->solrParams);
        $results = $solr->search($bcksolrQuery, $this->solrStart, $this->solrLimit, $this->solrParams);
    }
    catch (Exception $e)
    {
        drupal_set_message(t('error searching ') . $e->getMessage());
    }
    $this->solrResult = $results;
    if (empty($results))
    {
        drupal_set_message(t('Error searching solr index. Is the solr search block configured properly?'),
        'error');
        return;
    }
}

```

```

    }
    unset($results);
    unset($solr);
    return;
  }
  /**
   * Function delete_filter
   *
   * Deletes one filter from a multi-filter string
   *
   * @param <string> $facets
   * @param <string> $facet
   */
  function delete_filter($filters, $filter)
  {
    $filters = str_replace($filter, "", $filters);
    $filters = str_replace("~", "~", $filters);
    $filters = trim($filters, "~");
    return $filters;
  }
}

```

6.1.13 sites/all/modules/islandora_solr_search/islandora_solr_search.module

```

<?php
module_load_include('inc', 'islandora_solr_ingpiubook.shsearch', 'islandora_solr_search.admin');
module_load_include('inc', 'islandora_solr_search', 'includes/common');
module_load_include('inc', 'islandora_solr_search', 'IslandoraSolrQueryProcessor');
//Islandora_query;
$queryClass;
/**
 * @file
 * Implementation of Solr search for the Islandora fedora_search module.
 */
/**
 * Implementation of hook_menu().
 */
function islandora_solr_search_menu()
{
  $items['islandora/solr/search'] = array(
    'page callback' => 'islandora_solr_search',
    'access arguments' => array(
      'view fedora collection'
    ), //depends of fedora repository view
    'type' => MENU_CALLBACK
  );
  $items['islandora/solr/process'] = array(
    'page callback' => 'update_solr_url_div',
    'access arguments' => array(
      'view fedora collection'
    ),
    'file' => 'islandora_solr_search.admin.inc',
    'type' => MENU_CALLBACK
  );
  $items['admin/settings/islandora_solr_search'] = array(
    'title' => 'Islandora Solr Client',
    'description' => 'Managing Islandora Solr Searching',
    'page callback' => 'drupal_get_form',
    'page arguments' => array(
      'islandora_solr_admin_settings'
    ),
    'access arguments' => array(
      'administer site configuration'
    ),
    'file' => 'islandora_solr_search.admin.inc',
    'type' => MENU_NORMAL_ITEM
  );
  return $items;
}
/**
 * Implementation of hook_block().
 */
function islandora_solr_search_block($op = 'list', $delta = 0, $edit = array())
{
  global $queryClass;
  islandora_solr_search_init();
  /*
   * Here I need to call a hook which will return module/file/class/method/name/description
   * for blocks which need a queryClass to render.
   */
  $solr_blocks = module_invoke_all("islandora_solr_query_blocks");
  // The $op parameter determines what piece of information is being requested.
  switch ($op)
  {
    case 'list':
      // If $op is "list", we just need to return a list of block descriptions.
      // This is used to provide a list of possible blocks to the administrator,
      // end users will not see these descriptions.
      foreach ($solr_blocks as $name => $block)
      {
        $blocks[$name] = array(
          'info' => t($block['name'])
        );
      }
      return $blocks;
    case 'configure':
      // If $op is "configure", we need to provide the administrator with a

```

```

// configuration form. The $delta parameter tells us which block is being
// configured. In this example, we'll allow the administrator to customize
// the text of the first block.
$form = array();
return $form;
case 'save':
// If $op is "save", we need to save settings from the configuration form.
// Since the first block is the only one that allows configuration, we
// need to check $delta to make sure we only save it.
case 'view':
default:
// If $op is "view", then we need to generate the block for display
// purposes. The $delta parameter tells us which block is being requested.
if (!empty($solr_blocks[$delta]))
{
// First we'll set the block title.
$block['subject'] = t($solr_blocks[$delta]['name']);
// Include the file from which the block originates.
require_once(drupal_get_path('module', $solr_blocks[$delta]['module']) . '/' . $solr_blocks[$delta]
['file']);
// If a class is present, instantiate it and proceed from there.
// The variable $queryClass (the IslandoraSolrQueryProcessor, containing
// the Solr search result), is fed as an argument.
if (!empty($solr_blocks[$delta]['class']))
{
$displayClass = new IslandoraSolrResults();
$block_function = $solr_blocks[$delta]['function'];
if (method_exists($displayClass, $block_function))
{
$block['content'] = $displayClass->$block_function($queryClass);
}
// Otherwise, simply load the form.
}
elseif (!empty($solr_blocks[$delta]['form']))
{
$block['content'] = drupal_get_form($solr_blocks[$delta]['form']);
}
}
return $block;
}
}
/**
 * Implementation of hook_theme().
 */
function islandora_solr_search_theme()
{
return array(
'islandora_solr_search_block_form' => array(
'arguments' => array(
'form' => NULL
)
)
);
}
/**
 * Implements hook_islandora_solr_primary_display()
 */
function islandora_solr_search_islandora_solr_primary_display()
{
return array(
// 'machine-name' = array(
// 'name' => 'Human Readable Name',
// 'module' => 'module_name',
// 'file' => 'FileName.inc',
// 'class' => 'ClassName',
// 'function' => 'function_name',
// 'description' => 'A description of the display profile',
// );
//
// Note: this class should either be, or extend, the class IslandoraSolrResults.
//
'default' => array(
'name' => t('Fields (default)'),
'module' => 'islandora_solr_search',
'file' => 'IslandoraSolrResults.inc',
'class' => "IslandoraSolrResults",
'function' => "displayResults",
'description' => t("A simple output.")
)
);
}
/**
 * Implements hook_islandora_solr_query_blocks()
 */
function islandora_solr_search_islandora_solr_query_blocks()
{
return array(
// 'machine name' => array(
// 'name' => 'Human Readable Name',
// 'module' => 'module_name',
// 'file' => 'FileName.inc',
// 'class' => 'ClassName',
// 'function' => 'method name',
// 'form' => 'form_function_name',
// ),
//ingpiubook.sh

```

```

// Note: As in the examples below, it is valid to specify
// *either* a class and method *or* a form. The latter implies no
// class needs to be instantiated.
//
'advanced' => array(
  'name' => 'Islandora Solr Advanced Search',
  'module' => 'islandora_solr_search',
  'file' => 'islandora_solr_search.module',
  'class' => null,
  'function' => null,
  'form' => 'islandora_solr_search_block_form'
),
'simple' => array(
  'name' => 'Islandora Solr Simple Search',
  'module' => 'islandora_solr_search',
  'file' => 'islandora_solr_search.module',
  'class' => null,
  'function' => null,
  'form' => 'islandora_solr_simple_search_form'
),
'basic_facets' => array(
  'name' => 'Islandora Facet Block',
  'module' => 'islandora_solr_search',
  'file' => 'IslandoraSolrResults.inc',
  'class' => 'IslandoraSolrResults',
  'function' => 'displayFacets',
  'form' => null
),
'current_query' => array(
  'name' => 'Islandora Current Query',
  'module' => 'islandora_solr_search',
  'file' => 'IslandoraSolrResults.inc',
  'class' => 'IslandoraSolrResults',
  'function' => 'currentQuery',
  'form' => nullingpiubook.sh
)
);
/**
 * islandora solr simple search form
 * @return type
 */
function islandora_solr_simple_search_form()
{
  module_load_include('inc', 'islandora_solr_search', 'IslandoraSolrResults');
  $resultsClass = new IslandoraSolrResults();
  return $resultsClass->build_simple_solr_form();
}
/**
 * islandora solr search block form validate
 * @param type $form
 * @param type $form_state
 */
function islandora_solr_search_block_form_validate($form, &$form_state)
{
  $repeat = variable_get('islandora_solr_search_block_repeat', '3');
  $found = FALSE;
  for ($fieldNum = 1; $fieldNum <= $repeat; $fieldNum++)
  {
    if (isset($form_state['values']["fedora_terms$fieldNum"]))
    {
      $found = TRUE;
    }
  }
  if (!$found)
  {
    form_set_error('edit_fedora_terms1', t('Please enter search term.));
  }
}
/**
 * islandora solr simple search form submit
 * @param type $form
 * @param type $form_state
 */
function islandora_solr_simple_search_form_submit($form, &$form_state)
{
  $searchString = $form_state['values']['islandora_simple_search_query'];
  $searchString = htmlspecialchars(drupal_urlencode($searchString), ENT_QUOTES, 'utf-8', false);
  $searchString = str_replace('/', '~slash-', $searchString); //replace the slash so url doesn't break
  if (variable_get('dismax_allowed', FALSE))
  {
    $dismax = 'dismax';
  }
  drupal_goto("islandora/solr/search/$searchString/-/$dismax");
}
/**
 * islandora solr search block form
 * @global type $queryClass
 * @return type
 */
function islandora_solr_search_block_form()
{
  global $queryClass;
  islandora_solr_search_init();
  module_load_include('inc', 'islandora_solr_search', 'IslandoraSolrResults');
  $resultsClass = new IslandoraSolrResults();

```

```

return $resultsClass->build_solr_search_form(null, null, $queryClass->solrQuery);
}
/**
 * theme islandora solr search block form
 * @global type $queryClass
 * @param type $form
 * @return type
 */
function theme_islandora_solr_search_block_form($form)
{
    global $queryClass;
    islandora_solr_search_init();
    module_load_include('inc', 'islandora_solr_search', 'IslandoraSolrResults');
    $resultsClass = new IslandoraSolrResults();
    return $resultsClass->theme_solr_search_form($form);
}
/**ingpiubook.sh
 * islandora solr search
 * @global type $queryClass
 * @param type $query
 * @param type $fq
 * @param type $dismax
 * @return type
 */
function islandora_solr_search($query, $fq = null, $dismax = null)
{
    global $queryClass;
    islandora_solr_search_init();
    // Build and execute Apache Solr query
    $queryResult = $queryClass->buildAndExecuteQuery($query, $fq, $dismax);
    // Get the preferred display profile
    // Order: First choice is what's in the ?profile query var
    // Second choice is the primary display profile
    // Third choice is the default IslandoraSolrResults
    $islandora_solr_primary_display = variable_get('islandora_solr_primary_display', 'default');
    $islandora_solr_selected_display = isset($_GET['solr_profile']) ? $_GET['solr_profile'] : null;
    $primary_profiles = module_invoke_all("islandora_solr_primary_display");
    $secondary_profiles = module_invoke_all("islandora_solr_secondary_display");
    // TODO: Also filter secondary displays against those checked in the configuration options.
    if (isset($secondary_profiles[$islandora_solr_selected_display]))
    {
        $profile = $secondary_profiles[$islandora_solr_selected_display];
    }
    elseif (isset($primary_profiles[$islandora_solr_primary_display]))
    {
        $profile = $primary_profiles[$islandora_solr_primary_display];
    }
    else
    {
        drupal_set_message("There is an error in the solr search configuration: the display profile is not found.",
        'error');
        $profile = $primary_profiles['default'];
    }
    // Include the file for the display profile
    require_once(drupal_get_path('module', $profile['module']) . '/' . $profile['file']);
    // Set display class and function vars
    $solrClass = $profile['class'];
    $solrFunction = $profile['function'];
    $nope = false;
    if (class_exists($solrClass))
    {
        $implementation = new $solrClass();
    }
    else
    {
        $nope = true;
    }
    if (!$nope && method_exists($implementation, $solrFunction))
    {
        $output = $implementation->$solrFunction($queryClass);
    }
    else
    {
        $nope = true;
    }
    if ($nope)
    {
        module_load_include('inc', 'islandora_solr_search', 'IslandoraSolrResults');
        $resultsClass = new IslandoraSolrResults();
        $output = $resultsClass->displayResults($queryClass);
    }
    if (variable_get('islandora_solr_search_debug_mode', 0)) // debug dump
    {
        drupal_set_message('Params: <br/><pre>' . print_r($queryClass->solrParams, true) . '</pre>', 'status');
    }
    return $output;
    // return $queryResult;
}
/**
 * islandora solr search block form submit
 * @global type $queryClass
 * @param type $form
 * @param type $form_state
 */
function islandora_solr_search_block_form_submit($form, &$form_state)
{

```

```

global $queryClass;
islandora_solr_search_init();
$type_id = trim($form_state['values']['type']);
$repeat = variable_get('islandora_solr_search_block_repeat', '3');
$fedora_terms = array();
$types = array();
$booleans = array();
for ($fieldNum = 1; $fieldNum <= $repeat; $fieldNum++)
{
  ingpiubook.sh
  if ($form_state['values']["fedora_terms$fieldNum"])
  {
    $types[] = trim($form_state['values']["type$fieldNum"]);
    $fedora_terms[] = lesser_escape(trim($form_state['values']["fedora_terms$fieldNum"]));
  }
  $next = $fieldNum + 1;
  if ($form_state['values']["fedora_terms$next"] && $form_state['values']["fedora_terms$fieldNum"])
  {
    $booleans[] = trim($form_state['values']["andor$fieldNum"]);
  }
}
for ($iteration = 0; $iteration < count($fedora_terms); $iteration++)
{
  $searchString .= $types[$iteration] . ":" . $fedora_terms[$iteration] . " " . $booleans[$iteration] . " ";
}
// mando sempre AND o OR alla fine dopo due punti
$searchString .= ":" . trim($form_state['values']["andor0"]);
//
$searchString = trim($searchString);
$searchString = htmlspecialchars(drupal_urlencode($searchString), ENT_QUOTES, 'utf-8', false);
$searchString = str_replace('/', '~slsh~', $searchString); //replace the slash so url doesn't break
drupal_goto("islandora/solr/search/$searchString/-/");
}
/**
 * Implementation of hook_help().
 */
function islandora_solr_search_help($path, $arg)
{
  switch ($path)
  {
    case 'admin/help#islandora_solr_search':
      return t('<p>
The Islandora Solr Search extends the functionality of the Fedora_Repository module.
This module allows one or more of a series of blocks to be configured to search a solr index.
This module can co-exist with the original Fedora_Repositories search block, but Solr\'s
additional functionality will normally make the original block redundant.
</p>
<p>
The !guide contains additional information.
</p>
<ul>
<li>Islandora Solr Search requires a working Solr instance. The !sWiki has full setup
instructions.</li>
<li>Once Solr is running and tested, configure <b>Gsearch</b> to update Solr. Consult the !GSearch for
details.</li>
<li>Retreive the !client, unzip it, and copy the <b>Solr</b> directory from the archive to the
islandora_solr_search module\'s folder.</li>
<li>Go to Administer &gt; Site Configuration &gt; Islandora Solr Client <em>(or click the link
below)</em> to configure the module. Set which Solr request handler to use, set the port, host and context for
the index to be queried, and select which fields are to be used for filtering. Solr\'s <b>schema.xml</b> and
<b>solrconfig.xml</b> must be configured for the request handler as well as which fields to index and
return.</li>
<li>The module allows custom code to be used to display search results.&nbsp; If custom PHP code is
used, the paths to that codes\'s file and function must be entered here as well.</li>
<li>Three different blocks are now available under Administer &gt; Site Building &gt; Blocks:&nbsp;
Islandora Solr Simple Search Block, Islandora Solr Facet Block, and Islandora Solr Search Block.&nbsp; The
configuration setting for each of these blocks will allow you to control their position on the screen, and on
which pages they will be displayed.</li>
<li>The Islandora Solr Simple Search Block will use will add defType=dismax to the configured request
handler. The request handler tag in <b>solrconfig.xml</b> must have an attribute of
<b>default="true"</b>.</li>
</ul>
', array(
  '!guide' => l('Islandora Guide', 'https://wiki.duraspace.org/display/ISLANDORA/Islandora+Guide'),
  '!sWiki' => l("Solr Wiki", 'http://wiki.apache.org/solr/SolrTomcat'),
  '!GSearch' => l('GSearch Documentation',
'https://wiki.duraspace.org/display/FCSVCS/Generic+Search+Service+2.2'),
  '!client' => l('Apache Solr php client', 'http://code.google.com/p/solr-php-client')
));
}
}
/**
 * islandora_solr_search_init
 * @global IslandoraSolrQueryProcessor $queryClass
 * @staticvar boolean $islandora_solr_search_init
 */
function islandora_solr_search_init()
{
  static $islandora_solr_search_init = false;
  if (!$islandora_solr_search_init)
  {
    drupal_add_css(drupal_get_path('module', 'islandora_solr_search') . '/islandora_solr_search.css');
    global $queryClass;
    if (empty($queryClass))
    {
      $queryClass = new IslandoraSolrQueryProcessor();
    }
  }
}

```

```

        $islandora_solr_search_init = true;
    }
}

```

6.2 Back-end server

6.2.1 ingpiubook.sh

```

#!/bin/bash

FEDORA_HOME="/usr/local/fedora"
export FEDORA_HOME

PATH=$PATH:$FEDORA_HOME/server/bin:$FEDORA_HOME/client/bin:$JAVA_HOME/bin:/bin:/usr/bin:/sbin:/usr/sbin
export PATH

# $1: directory with book directories
# example: IMPORT_BASE_DIR="/srv/storage/scansioni/cognetti_collezioni/direttori/"
IMPORT_BASE_DIR=$1

# $2: collection PID
# example: IMPORT_COLLECTION="openbess:cognetti-C001"
IMPORT_COLLECTION=$2

# $3: admin password or 'test'

# $4: number of pages to ingest or 0 for all

n=0
for ndir in $(find "$IMPORT_BASE_DIR"* -maxdepth 0 -type d);
do
    if [[ "$3" != test ]]
    then
        sleep 60
    else
        echo "NO wait"
    fi

    ./ingbookepages.sh "$ndir" $2 $3 $4
done

```

6.2.2 ingbookepages.sh

```

#!/bin/bash

FEDORA_HOME="/usr/local/fedora"
export FEDORA_HOME

PATH=$PATH:$FEDORA_HOME/server/bin:$FEDORA_HOME/client/bin:$JAVA_HOME/bin:/bin:/usr/bin:/sbin:/usr/sbin
export PATH

# 1 Book directory ? e.g. "/srv/storage/fgramsci_T0024-00001"
IMPORT_BASE_DIR=$1
# 2 Collection PID ? e.g. "openbess:cognetti-C001"
COLLEZIONE=$2
# 3 fedoraAdmin password OR "test"
# 4 number of pages OR "0" for all

IMPORT_VOL_DIR=${IMPORT_BASE_DIR##*/}
bookPID=${IMPORT_VOL_DIR/" /":}
imagedir="${IMPORT_BASE_DIR}"/

echo "=====
echo "Book PID: $bookPID"
echo "dir: $imagedir"
echo "=====
#
## MAIN CHECK ##
#
# check TIFF files #
SAVEIFS=$IFS
IFS=$(echo -en "\n\b")
ntiff=0
for nfile in $(find "$imagedir"*.tif -type f);
do
    let "ntiff += 1"
    sn=$(printf "%04d" $ntiff)
    if [[ "$nfile" != *$sn.tif ]]
    then
        echo "ERROR file $nfile NotEqual $sn"
        exit
    fi
done
IFS=$SAVEIFS
#
# check subdirectory #
n=0
for ndir in $(find "$imagedir"* -type d);
do
    let "n += 1"
done
if [ $n -gt 1 ] || [ $n -lt 1 ]

```

```

then
    echo "ERROR subdirectory NotUnique OR NotExist"
    exit
fi
ocrdir=$(find "$imagedir"* -type d)
#
# check TXT files #
ntxt=0
SAVEIFS=$IFS
IFS=$(echo -en "\n\b")
for nfile in $(find "$ocrdir"/*.txt -type f);
do
    let "ntxt += 1"
    st=$(printf "%04d" $ntxt)
    if [[ "$nfile" != *$st.txt ]]
    then
        echo "ERROR file $nfile NotEqual $st"
        exit
    fi
done
IFS=$SAVEIFS
#
# check TIFF and TXT number of files #
if [ $ntiff -ne $ntxt ]
then
    echo "ERROR files $ntiff TIFF NotEqual $ntxt TXT"
    exit
fi
#
# check PDF file #
n=0
SAVEIFS=$IFS
IFS=$(echo -en "\n\b")
for nfile in $(find "$imagedir"*.pdf -type f);
do
    let "n += 1"
    filepdf="$nfile"
done
if [ $n -gt 1 ] || [ $n -lt 1 ]
then
    echo "ERROR file PDF NotUnique OR NotExist"
exit
fi
IFS=$SAVEIFS
cp $filepdf "/srv/temp/pdf.pdf"
filepdf="/srv/temp/pdf.pdf"
#
## DC INDEX CHECK, EXTRACT ##
#
# check file DCIndex #
n=0
SAVEIFS=$IFS
IFS=$(echo -en "\n\b")
for nfile in $(find "$imagedir"*.txt -type f);
do
    let "n += 1"
    filedcindice="$nfile"
done
if [ $n -gt 1 ] || [ $n -lt 1 ]
then
    echo "ERROR file DCIndex NotUnique OR NotExist"
    exit
fi
IFS=$SAVEIFS
#substitute problematic chars
sed -i s/\?/\!/g $filedcindice
sed -i s/\[/\(/g $filedcindice
sed -i s/\]/\)/g $filedcindice
cp "$filedcindice" "/srv/temp/fdci.dci"
chmod +w "/srv/temp/fdci.dci"
#convert from dos to unix
fromdos "/srv/temp/fdci.dci"
filedcindice="/srv/temp/fdci.dci"
#check exist
if [[ ! -e $filedcindice ] || [ ! -f $filedcindice ] || [ ! -s $filedcindice ]
then
    if [[ "$3" != test ]]
    then
        exit
    else
        echo "ERROR file $filedcindice"
    fi
fi
#check utf-8 or us-ascii
filetype=$(file -bi $filedcindice)
if [[ "$filetype" != *utf-8 ] ] && [[ "$filetype" != *us-ascii ] ]
then
    if [[ "$3" != test ]]
    then
        exit
    else
        echo "ERROR file $filedcindice $filetype"
    fi
fi
#remove first 3 special chars if exist
awk '{if(NR=1)sub(/^[\xef\xbb\xbf/, ""]);print}' "/srv/temp/fdci.dci" > "/srv/temp/fdci.idx"

```

```

rm "/srv/temp/fdci.dci"
filedcindice="/srv/temp/fdci.idx"
#
# DC check and extract to array #
DCN=( [DCTITLE] [DCCREATOR] [DCSUBJECT] [DCDESCRIPTION] [DCPUBLISHER] [DCDATE] [DCTYPE] [DCFORMAT] [DCLANGUAGE] )
DCV[0]=""
linind=""
ndc=0
nind=0
while read line; do
    lline=${#line}
    if [ $lline -gt 0 ]
    then
        dcvar="${line%%:*}"
        dcval="${line#*:}"
        pc=${line:0:1}
        if [[ "$pc" < "0" ]] || [[ "$pc" > "9" ]]
        then
            let "ndc += 1"
            case $dcvar in
                TITLE)
                    DCV[0]="$dcval"
                    ;;
                CREATOR)
                    DCV[1]="$dcval"
                    ;;
                SUBJECT)
                    DCV[2]="$dcval"
                    ;;
                DESCRIPTION)
                    DCV[3]="$dcval"
                    ;;
                PUBLISHER)
                    DCV[4]="$dcval"
                    ;;
                DATE)
                    case $dcval in
                        [0-9][0-9][0-9][0-9]-[0-9]-[0-9][0-9])
                            DCV[5]="$dcval"
                            ;;
                        [0-9][0-9][0-9][0-9][0-9]-[0-9][0-9])
                            DCV[5]="$dcval"
                            ;;
                        [0-9][0-9][0-9][0-9])
                            DCV[5]="$dcval"
                            ;;
                        *)
                            echo "ERROR DC Date Format: $dcval"
                            exit
                            ;;
                    esac
                ;;
                TYPE)
                    DCV[6]="$dcval"
                    ;;
                FORMAT)
                    DCV[7]="$dcval"
                    ;;
                LANGUAGE)
                    DCV[8]="$dcval"
                    ;;
                *)
                    echo "ERROR DC metadata: $dcvar val: $dcval"
                    exit
                    ;;
            esac
        fi
    fi
done < "$filedcindice"
#
# check DC TITLE #
ldctitle=${#DCV[0]}
if [ $ldctitle -gt 0 ]
then
    linkind="$bookPID|${DCV[0]}"
else
    echo "ERROR DC TITLE NotExist"
    exit
fi
#
# show DC metadata #
for i in {0..8}
do
    echo "${DCN[$i]} = ${DCV[$i]}"
done
#
# check Index #
while read line; do
    lline=${#line}
    if [ $lline -gt 0 ]
    then
        dcvar="${line%%:*}"
        dcval="${line#*:}"
        pc=${line:0:1}
        if !([[ "$pc" < "0" ]] || [[ "$pc" > "9" ]])
        then

```

```

        case $dcvar in
        [0-9][0-9][0-9][0-9])
            echo "$dcval" page "$dcvar"
            let "nind += 1"
            linkind="$linkind"|$bookPID-$dcvar|$dcval"
            ;;
        *)
            echo "ERROR Index: $dcval page $dcvar"
            exit
            ;;
        esac
    fi
done < "$filedcindice"
rm "$filedcindice"
#
# create datastream INDEX file #
fileindex="/srv/temp/idx.idx"
if [ -e $fileindex ]; then
    rm $fileindex
fi
echo "$linkind" > $fileindex
#
## BOOK ##
#
# create thumbnail image from TIFF 0001 #
IFS=$(echo -en "\n\b")
filetiff=$(find "$imagedir"*0001.tif -type f)
IFS=$SAVEIFS
if [ ! -e $filetiff ];
then
    echo "ERROR file TIFF 0001 for TN $filetiff"
    exit
fi
#remove tiff compression
tiffcp -c none "$filetiff" "/srv/temp/tiff.tmp"
#don't convert to gray scale
#convert "/srv/temp/tiff.tmp" -colorspace Gray "/srv/temp/tiff.tif"
rm "/srv/temp/tiff.tmp"
filetn="/srv/temp/tnbook.jpg"
#resize image
risconvert=$(convert "/srv/temp/tiff.tif"[0] -thumbnail 200x225 -fuzz 1% +repage -gravity center -format jpg
-quality 100 "$filetn")
if [ ! -e $filetn ] || [ ! -f $filetn ] || [ ! -s $filetn ]
then
    echo "ERROR file $filetn : $risconvert"
    exit
fi
risconvert=$(convert "$filetn" -colorspace RGB "$filetn")
if [ ! -e $filetn ] || [ ! -f $filetn ] || [ ! -s $filetn ]
then
    echo "ERROR file $filetn : $risconvert"
    exit
fi
#
# prepare book template #
cp import/templateBookMaster.xml import/templateBook.xml
pidlabel=${DCV[0]}
pidlabel=${DCV[0]//"/"/"\}
pidlabel=${pidlabel//"/"/"\&}
pidlabel=${pidlabel:0:250}
sed -i "s/[PID]/$bookPID/g;s/[PIDLABEL]/$pidlabel/g;s/[COLLEZIONE]/$COLLEZIONE/g" import/templateBook.xml
for i in {0..8}
do
    var=${DCN[$i]//"/"/"\}
    var=${var//"/"/"\}
    valo=${DCV[$i]//"/"/"\}
    valo=${valo//"/"/"\&}
    sed -i "s/$var/$valo/g" import/templateBook.xml
done
ftn=${filetn//"/"/"\}
findex=${fileindex//"/"/"\}
fpdf=${filepdf//"/"/"\}
sed -i "s/[FILETN]/$ftn/g;s/[FILEINDEX]/$findex/g;s/[FILEPDF]/$fpdf/g" import/templateBook.xml
#
# ingest book or nothing if test #
if [[ "$3" != test ]]
then
    risultato=$(/usr/local/fedora/client/bin/fedora-ingest.sh f import/templateBook.xml info:fedora/fedora-
system:FOXML-1.1 fcl.to.cnr.it:8080 fedoraAdmin $3 http)
    if [[ "$risultato" == Ingested* ]]
    then
        echo "$risultato"
    else
        echo "ERROR INGESTING $risultato"
        exit
    fi
else
    echo "-> INGESTED(test)"
fi
if [[ "$3" != test ]]
then
    sleep 120
else
    echo "NO wait"
fi

```

```

#
## PAGES ##
#
# evaluate number of pages to ingest #
if [ $4 -gt 0 ]
then
    finoa=$4
else
    finoa=ntiff
fi
# cycle single page #
for (( npage=1; npage<=finoa; npage++ ))
do
#
# define PID and Label #
snpage=$(printf "%04d" $npage)
pagePID="$bookPID"-"$snpage"
valo=${DCV[0]//"/"/"\\"}
valo=${valo//"/"/"\&"}
pageDCTitle="$valo" - page "$snpage"
pagePIDlabel=${valo:0:240} - page "$snpage"

#
# check tiff #
IFS=$(echo -en "\n\b")
filetiff=$(find "$imagedir"*$snpage.tif -type f)
IFS=$SAVEIFS
if [ ! -e $filetiff ] || [ ! -f $filetiff ] || [ ! -s $filetiff ]
then
    if [[ "$3" != test ]]
    then
        exit
    else
        echo "$snpage -> ERROR file $filetiff"
    fi
fi

#
# remove tiff compression #
tiffcp -c none "$filetiff" "/srv/temp/tiff.tmp"

#
# convert tiff to gray scale #
convert "/srv/temp/tiff.tmp" -colorspace Gray "/srv/temp/tiff.tif"
rm "/srv/temp/tiff.tmp"
filetiff="/srv/temp/tiff.tif"

#
# create page thumbnail #
filetn="/srv/temp/page-tn.jpg"
risconvert=$(convert "$filetiff"[0] -thumbnail 200x225 -fuzz 1% +repage -gravity center -format jpg
-quality 100 "$filetn")
if [ ! -e $filetn ] || [ ! -f $filetn ] || [ ! -s $filetn ]
then
    if [[ "$3" != test ]]
    then
        exit
    else
        echo "$snpage -> ERROR file $filetn : $risconvert"
    fi
fi

#
# create page jpeg2000 #
filej2="/srv/temp/jp2.jp2"
riscompres=$(kdu_compress -i "$filetiff" -o "$filej2" -rate 0.5 Clayers=1 Clevels=7
"Cpcreincts={256,256},{256,256},{128,128},{128,128},{128,128},{64,64},{64,64},{32,32},{16,16}" "Corder=RPCL"
"ORGgen plt=yes" "ORGtparts=R" "Cblk={32,32}" Cuse_sop=yes)
if [ ! -e $filej2 ] || [ ! -f $filej2 ] || [ ! -s $filej2 ]
then
    if [[ "$3" != test ]]
    then
        exit
    else
        echo "$snpage -> ERROR file $filej2 : $riscompres"
    fi
fi

#
# prepare txt file #
IFS=$(echo -en "\n\b")
fileocr=$(find "$ocdir"/*$snpage.txt -type f)
IFS=$SAVEIFS
#remove char FF &#12 oct 014
tr -d '\014' < "$fileocr" > "/srv/temp/ocr.tmp"
#convert from dos to unix
fromdos "/srv/temp/ocr.tmp"
fileocrtmp="/srv/temp/ocr.tmp"
#check exist
if [ ! -e $fileocrtmp ] || [ ! -f $fileocrtmp ]
then
    if [[ "$3" != test ]]
    then
        exit
    else
        echo "$snpage -> ERROR file $fileocrtmp"
    fi
fi

filel=$(stat -c %s $fileocrtmp)
filenull=4
#check zero lenght
if [ ! -s $fileocrtmp ] || [ "$filel" -le "$filenull" ]

```

```

then
    cp blank.txt $fileocrtmp
    echo "blank file OCR"
fi
#check utf-8 or us-ascii
filetype=$(file -bi $fileocrtmp)
if [[ "$filetype" != *utf-8 ]] && [[ "$filetype" != *us-ascii ]]
then
    if [[ "$3" != test ]]
    then
        exit
    else
        echo "$snpage -> ERROR file $fileocrtmp"
    fi
fi
#remove first 3 special chars if exist
awk '{if(NR==1)sub(/^\xef\xbb\xbf/,"");print}' "/srv/temp/ocr.tmp" > "/srv/temp/ocr.ocr"
rm "/srv/temp/ocr.tmp"
fileocrtmp="/srv/temp/ocr.ocr"
#
# prepare page template #
cp import/templatePageMaster.xml import/templatePage.xml
sed -i "s/\[PID\]/$pagePID/g;s/\[PIDLABEL\]/$pagePIDlabel/g" import/templatePage.xml
sed -i "s/\[DCTITLE\]/$pageDCTitle/g;s/\[PIDBOOK\]/$bookPID/g" import/templatePage.xml
for i in {1..8}
do
    var=${DCN[$i]//"/"/"}
    var=${var//"/"/"}
    valo=${DCV[$i]//"/"/"}
    valo=${valo//"/"/}
    sed -i "s/$var/$valo/g" import/templatePage.xml
done
ftn=${filetn//"/"/}
fjp2=${filejp2//"/"/}
focr=${fileocrtmp//"/"/}
ftiff=${filetiff//"/"/}
sed -i "s/\[FILETN\]/$ftn/g;s/\[FILEJP2\]/$fjp2/g;s/\[FILEOCR\]/$focr/g" import/templatePage.xml
sed -i "s/\[FILETIFF\]/$ftiff/g" import/templatePage.xml
#
# ingest page or nothing if test #
if [[ "$3" != test ]]
then
    risultato=$(/usr/local/fedora/client/bin/fedora-ingest.sh f import/templatePage.xml
info:fedora/fedora-system:FOXML-1.1 fcl.to.cnr.it:8080 fedoraAdmin $3 http)
    if [[ "$risultato" == Ingested* ]]
    then
        echo "$snpage -> $risultato"
    else
        echo "$snpage -> ERROR INGESTING $risultato"
        exit
    fi
else
    echo "$snpage -> INGESTED(test)"
fi
done
exit

```

6.2.3 templateBookMaster.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<foxml:digitalObject xmlns:foxml="info:fedora/fedora-system:def/foxml#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" VERSION="1.1" PID="[PID]"
xsi:schemaLocation="info:fedora/fedora-system:def/foxml# http://www.fedora.info/definitions/1/0/foxml1-1.xsd">
<foxml:objectProperties>
<foxml:property NAME="info:fedora/fedora-system:def/model#state" VALUE="A"/>
<foxml:property NAME="info:fedora/fedora-system:def/model#label" VALUE="[PIDLABEL]"/>
<foxml:property NAME="info:fedora/fedora-system:def/model#ownerId" VALUE="fedoraAdmin"/>
</foxml:objectProperties>
<foxml:datastream ID="DC" STATE="A" CONTROL_GROUP="X">
<foxml:datastreamVersion FORMAT_URI="http://www.openarchives.org/OAI/2.0/oai_dc/" ID="DC.0"
MIMETYPE="text/xml" LABEL="Dublin Core Record for this object">
<foxml:xmlContent>
<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
xmlns:dc="http://purl.org/dc/elements/1.1/">
<dc:title>[DCTITLE]</dc:title>
<dc:creator>[DCCREATOR]</dc:creator>
<dc:subject>[DCSUBJECT]</dc:subject>
<dc:description>
[DCDESCRIPTION]
</dc:description>
<dc:type>[DCTYPE]</dc:type>
<dc:date>[DCDATE]</dc:date>
<dc:publisher>[DCPUBLISHER]</dc:publisher>
<dc:format>[DCFORMAT]</dc:format>
<dc:language>[DCLANGUAGE]</dc:language>
<dc:identifier>[PID]</dc:identifier>
<dc:identifier>http://www.digibess.it/fedora/repository/[PID]</dc:identifier>
</oai_dc:dc>
</foxml:xmlContent>
</foxml:datastreamVersion>
</foxml:datastream>
<foxml:datastream ID="RELS-EXT" CONTROL_GROUP="X">
<foxml:datastreamVersion FORMAT_URI="info:fedora/fedora-system:FedoraRELSExt-1.0" ID="RELS-EXT.0"
MIMETYPE="application/rdf+xml" LABEL="RDF Statements about this object">
<foxml:xmlContent>
<rdf:RDF xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:fedora="info:fedora/fedora-

```

```

system:def/relations-external#" xmlns:fedora-model="info:fedora/fedora-system:def/model#"
xmlns:oai="http://www.openarchives.org/OAI/2.0/" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  <rdf:Description rdf:about="info:fedora/[PID]">
    <fedora:isMemberOfCollection rdf:resource="info:fedora/[COLLEZIONE]"/>
    <fedora-model:hasModel rdf:resource="info:fedora/islandora:bookCModel"/>
    <oai:itemID>oai:fcl.to.cnr.it:[PID]</oai:itemID>
  </rdf:Description>
</rdf:RDF>
</foxml:xmlContent>
</foxml:datastreamVersion>
</foxml:datastream>
<foxml:datastream CONTROL_GROUP="M" ID="TN" STATE="A">
  <foxml:datastreamVersion ID="TN.0" MIMETYPE="image/jpeg" LABEL="Thumbnail">
    <foxml:contentLocation REF="file:[FILETN]" TYPE="URL"/>
  </foxml:datastreamVersion>
</foxml:datastream>
<foxml:datastream CONTROL_GROUP="M" ID="INDEX" STATE="A">
  <foxml:datastreamVersion ID="INDEX.0" MIMETYPE="text/plain" LABEL="Index">
    <foxml:contentLocation REF="file:[FILEINDEX]" TYPE="URL"/>
  </foxml:datastreamVersion>
</foxml:datastream>
<foxml:datastream CONTROL_GROUP="M" ID="PDF" STATE="A">
  <foxml:datastreamVersion ID="PDF.0" MIMETYPE="application/pdf" LABEL="Reference PDF">
    <foxml:contentLocation REF="file:[FILEPDF]" TYPE="URL"/>
  </foxml:datastreamVersion>
</foxml:datastream>
</foxml:digitalObject>
    
```

6.2.4 templatePageMaster.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<foxml:digitalObject xmlns:foxml="info:fedora/fedora-system:def/foxml#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" VERSION="1.1" PID="[PID]"
xsi:schemaLocation="info:fedora/fedora-system:def/foxml# http://www.fedora.info/definitions/1/0/foxml1-1.xsd">
  <foxml:objectProperties>
    <foxml:property NAME="info:fedora/fedora-system:def/model#state" VALUE="A"/>
    <foxml:property NAME="info:fedora/fedora-system:def/model#label" VALUE="[PIDLABEL]"/>
    <foxml:property NAME="info:fedora/fedora-system:def/model#ownerId" VALUE="fedoraAdmin"/>
  </foxml:objectProperties>
  <foxml:datastream ID="DC" STATE="A" CONTROL_GROUP="X">
    <foxml:datastreamVersion FORMAT_URI="http://www.openarchives.org/OAI/2.0/oai_dc/" ID="DC.0"
MIMETYPE="text/xml" LABEL="Dublin Core Record for this object">
      <foxml:xmlContent>
        <oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
xmlns:dc="http://purl.org/dc/elements/1.1/">
          <dc:title>[DCTITLE]</dc:title>
          <dc:creator>[DCCREATOR]</dc:creator>
          <dc:subject>[DCSUBJECT]</dc:subject>
          <dc:description>
            [DCDESCRIPTION]
          </dc:description>
          <dc:type>[DCTYPE]</dc:type>
          <dc:date>[DCDATE]</dc:date>
          <dc:publisher>[DCPUBLISHER]</dc:publisher>
          <dc:format>[DCFORMAT]</dc:format>
          <dc:language>[DCLANGUAGE]</dc:language>
          <dc:identifier>[PID]</dc:identifier>
        </oai_dc:dc>
      </foxml:xmlContent>
    </foxml:datastreamVersion>
  </foxml:datastream>
  <foxml:datastream ID="RELS-EXT" CONTROL_GROUP="X">
    <foxml:datastreamVersion FORMAT_URI="info:fedora/fedora-system:FedoraRELSExt-1.0" ID="RELS-EXT.0"
MIMETYPE="application/rdf+xml" LABEL="RDF Statements about this object">
      <foxml:xmlContent>
        <rdf:RDF xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:fedora="info:fedora/fedora-
system:def/relations-external#" xmlns:fedora-model="info:fedora/fedora-system:def/model#"
xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
          <rdf:Description rdf:about="info:fedora/[PID]">
            <fedora:isMemberOf rdf:resource="info:fedora/[PIDBOOK]"/>
            <fedora-model:hasModel rdf:resource="info:fedora/islandora:pageCModel"/>
          </rdf:Description>
        </rdf:RDF>
      </foxml:xmlContent>
    </foxml:datastreamVersion>
  </foxml:datastream>
  <foxml:datastream CONTROL_GROUP="M" ID="TN" STATE="A">
    <foxml:datastreamVersion ID="TN.0" MIMETYPE="image/jpeg" LABEL="Thumbnail">
      <foxml:contentLocation REF="file:[FILETN]" TYPE="URL"/>
    </foxml:datastreamVersion>
  </foxml:datastream>
  <foxml:datastream CONTROL_GROUP="M" ID="JP2" STATE="A">
    <foxml:datastreamVersion ID="JP2.0" MIMETYPE="image/jp2" LABEL="Compressed JP2">
      <foxml:contentLocation REF="file:[FILEJP2]" TYPE="URL"/>
    </foxml:datastreamVersion>
  </foxml:datastream>
  <foxml:datastream CONTROL_GROUP="M" ID="OCR" STATE="A">
    <foxml:datastreamVersion ID="OCR.0" MIMETYPE="text/plain" LABEL="Scanned text">
      <foxml:contentLocation REF="file:[FILEOCR]" TYPE="URL"/>
    </foxml:datastreamVersion>
  </foxml:datastream>
  <foxml:datastream CONTROL_GROUP="M" ID="TIFF" STATE="A">
    
```

```
<foxml:datastreamVersion ID="TIFF.0" MIMETYPE="image/tiff" LABEL="Reference HD image">
  <foxml:contentLocation REF="file:[FILETIFF]" TYPE="URL"/>
</foxml:datastreamVersion>
</foxml:datastream>
</foxml:digitalObject>
```