ELAR Discovery Via VUFIND	
Report type:	VuFind Technical Documentation
Date:	14 April 2016
Author:	Josu Moreno
Department:	LMS and Library Services
Version:	1.2

Table of Content

1. SCOPE

This document aims to be a VuFind technical documentation for SOAS IT staff to provide a first level support service for the ELAR interface via VuFind implementation delivered by SCANBIT

2. SERVER INSTALLATION

SYSTEM REQUIREMENTS

Software requirements for VuFind 2.5 installation:

- Apache 2.2.12+
- PHP 5.4+

- MySQL 5.1.10+
- Java JDK 1.7+
- PostgreSQL 9.0+

Additionally, VuFind installation requires at least 2GB of memory RAM and 15 GB of hard disk.

Vufind components and description

- Apache is a web server. VuFind needs Apache so that its pages are visible to users who want to access them.
- Solr is a search engine. VuFind uses Solr to index your records and search through them for users. VuFind interacts with Solr in exactly the same way that your web browser talks to a web server. To make this possible, Solr runs inside its own web server software called Jetty.
- PHP is the programming language that was used to write VuFind. Apache uses PHP to turn VuFind's code into web pages customized to answer user requests. PHP is the engine that drives all of VuFind's interactivity; without it, VuFind wouldn't be able to do anything.
- An Integrated Library Management System (ILMS, ILS) is the software that traditionally handles catalog searches as well as circulation and administration. VuFind is designed to talk to an ILMS of some sort, though non-library users may be able to use it in other creative ways.
- MySQL is the Database Management System (DBMS) that houses VuFind's local application database for your social metadata and such. When users add tags or leave comments, that information is stored in the MySQL database.

All VuFind's web server configuration files are stored in /usr/local/vufind2 folder, this are the main configuration and script files

/usr/local/vufind2/local/httpd-vufind.conf . Apache configuration for linking with VuFind

/usr/local/vufind2/config/vufind/config.ini . Main configuration file of VuFind

/usr/local/vufind2/vufind.sh . Startup/shutdown script of VuFind daemon

VuFind additional documentation can be retrieved from official webpage https://vufind.org

VUFIND SERVERS FOR SOAS ELAR

Currently, there are 3 servers for the project ELAR discovery Via VuFind:

Development	http://golwg.lis.soas.ac.uk/
Test	http://sjon.lis.soas.ac.uk/
Production	http://wurin.lis.soas.ac.uk/

The **access to these servers** requires a username (person name) and, to perform the connection as root, it will be necessary to insert the 'su' command and the root password.

To **upload files** to those platforms, the user will access to the FTP/SSH using their person username. The files will be saved in the folder /home/xxxxx. After that, from the terminal and using root as username, those files will be moved to the desired path.

To create a **back up** of the MySQL database, there is a script in /root/scripts/ called backup_mysql.sh. This script creates a file containing the copy of the database within the path /usr/local/vufind/mysqldump/. That resulting file is named with the creation date. In addition to the database, it will also be necessary to copy the content of the folders /usr/local/vufind and /root/scripts (the harvesting and importing scripts used by the cron tasks are stored in thsis folder)

3. VUFIND FOR PUBLISHING LAT DATA VIA DISCOVERY INTERFACE

LAT is the application used by SOAS ELAR team to store, view and query language metadata and multimedia files. Before installing VuFind, the online public interface to search and retrieve references to those data and resources was based on Drupal open source content management system. VuFind was the discovery tool used by publishing the library catalog. Because of the more powerful search features of VuFind and because there was a short-term project in SOAS for grouping all the institution's resources under the same platform, the ELAR team decided to publish the catalogue of languages in a new VuFind installation.

The requirements for the new interface based on VuFind included maintaining all the functionalities and the 'look and feel' already implemented in the old Drupal interface (http://elar.soas.ac.uk/). In addition, the VuFind specific features were customized and added to the new ELAR interface.

This is a list of the main features included in the current ELAR interface for searching and discovering the metadata and multimedia resources collected and catalogued in LAT by the ELAR team:

- Responsive design
- Browse by collections
- Alphabetical browse
- List of deposits
- Map of deposits
- Carroussel showing the latest records added to the platform
- One search box (All fields, Title, Keywords, Language)
- Specific filters or facets for LAT data to refine the results
- Display of images and embedded HTML
- Suggestion of similar items from the detailed view of a record
- Search bundles within a deposit
- Access to the multimedia resources.
- Images, audios and videos embedded through plugins

- SQL access to LAT/AMS database (access permissions, depositor view, …)
- Multi-tier data management
- Registration and login
- Active sitemaps for indexing the references in Google

4. HARVESTING AND IMPORTING DATA

The latest records catalogued in LAT are nightly exported from LAT (as XML) and imported and indexed in VuFind. Therefore, records added to LAT will be available for searching the day after they were catalogued.

VuFind is a discovery tool, initially for libraries, that works as a metasearch engine for publishing and indexing all the references to the library resources (references or metadata from library bibliographic catalogue, digital library, archives, bibliographies, …). The search engine is based on Solr.

To import the data from LAT (the management system for cataloguing language archives, used by ELAR), VuFind uses the OAI-PMH protocol (https://www.openarchives.org/pmh/). The data are harvesting daily (there is a scheduled task in the crontab) and the metadata that VuFind gets are stored in VuFind as XML files. Those files will be indexed in VuFind to make the records searchable. The process is as follows:

1. The **record is created in LAT**
2. The **record is exposed by LAT** and identified with an OAI url (URI)
3. **VuFind harvests the new records created in LAT** (it takes into account the last harvesting date: last_harvest). Once the records are harvested, the last harvesting date will be automatically updated in VuFind. The configuration file for the harvesting is saved as /usr/local/vufind/harvest/oai.ini

   The description for each of the fields imported in VuFind is in the same file oai.ini. This file doesn't require any changes (only if the OAI url changes).

   To perform the harvesting, run the following:

   cd /usr/local/vufind/harvest

   /usr/bin/php harvest_oai.php ELAR

   This is a standard output:

   Processing ELAR...

   Autodetecting date granularity... found YYYY-MM-DD.

   Processing 100 records...

   Processing 100 records...

   Processing 3 records...

Completed without errors -- 1 source(s) processed.

This example means that 203 records were processed using the last harvesting date. This date is saved in the file /usr/local/vufind/local/harvest/ELAR/last_harvest.txt. If the user requires to harvest records from a specific date, the administrator will have to edit this file. If the user requires to get all the records, the administrator will remove this file (the system will not take into account that date if the file doesn't exist). When this process is finished, the last harvesting date will be always the current date.

Processed records are stored in the path /usr/local/vufind/local/harvest/ELAR

4. **VuFind uses a style sheet to convert the input metadata** (deposits or bundles) into a XML file containing the fields that will be indexed by Solr search engine. This style sheet is saved as: /usr/local/vufind/local/import/xsl/elar-scb.xslt. There is another style sheet to convert and manage the depositor profiles provided by ELAR: /usr/local/vufind/local/import/xsl/authors.xsl

5. The resulting **XML files will be imported** in VuFind. To import the records, it is necessary to run the following:

   cd /usr/local/vufind/harvest

   /usr/local/vufind/harvest/batch-import-xsl.sh ./ELAR/ ../import/elar-scb.properties

   The expected output will be something like this:

   Processing /usr/local/vufind2/local/harvest/./ELAR//1460040639_oai_soas_ac_uk_MPI194589.xml ...

   Successfully imported /usr/local/vufind2/local/harvest/./ELAR//1460040639_oai_soas_ac_uk_MPI194589.xml...

   Processing /usr/local/vufind2/local/harvest/./ELAR//1460040639_oai_soas_ac_uk_MPI43292.xml ...

   Successfully imported /usr/local/vufind2/local/harvest/./ELAR//1460040639_oai_soas_ac_uk_MPI43292.xml...

   Processing /usr/local/vufind2/local/harvest/./ELAR//1460040639_oai_soas_ac_uk_MPI666480.xml ...

   Successfully imported /usr/local/vufind2/local/harvest/./ELAR//1460040639_oai_soas_ac_uk_MPI666480.xml...

Processing /usr/local/vufind2/local/harvest/./ELAR//1460044136_oai_soas_ac_uk_MPI43292.xml ...

Successfully imported /usr/local/vufind2/local/harvest/./ELAR//1460044136_oai_soas_ac_uk_MPI43292.xml...

Optimizing index...

If the records have been processed successfully, they will be saved in the folder /usr/local/vufind/local/harvest/ELAR/processed/. If there are any error, the record will remain as it is.

Once those files are imported, it is necessary to extract their depositors:

find $VUFIND_HOME/local/harvest/ELAR/processed -name '.xml' | xargs mv -t $VUFIND_HOME/local/harvest/Authors/*

/usr/local/vufind/harvest/batch-import-xsl-auth.sh ./Authors/ ../import/authors.properties

Finally, the alphabetic browse will require to be updated with the latest records added to VuFind:

cd /usr/local/vufind

./index-alphabetic-browse.sh

All those tasks are automatized in a script: /root/scripts/import_vufind.sh (latest records) and /root/scripts/import_vufind_full.sh (all the records)

6. The **scheduled task** should be within the crontab. Is should be daily. An example: 00 00 * * * /root/scripts/import_vufind_full.sh > /dev/null 2>&

5. ACCESS TO LOCAL FILES

The first requirement for the new ELAR interface based on VuFind was to replicate the design of the old Drupal interface. Because of that, to display the deposit data it was necessary to include some the embedded HTML of Drupal in VuFind. They are mainly links to image, audio or video files. These files were saved as:

/mnt/ELAR_Deposit_Resources

/mnt/ELAR_Home_Page_Resources

An Apache configuration file was also created to set the properties of those files:

/usr/local/vufind/local/httpd-vufind.conf

Within this file there are 3 alias:

Alias /projects/ "/mnt/ELAR_Home_Page_Resources/projects/"

Alias /swf/ "/mnt/ELAR_Home_Page_Resources/"

Alias /depositStore/ "/mnt/ELAR_Home_Page_Resources/"

All the external files linked from the Drupal code used in VuFind will be under the folder called deposit.

6. ACCESS TO LAT-AMS DATABASE

VuFind requires connection to LAT-AMS database to get required data not included in the metadata records that are imported and indexed in VuFind. The database has two instances: soas and corpustructure.

The configuration file to the database connection is stored as:

/usr/local/vufind/local/config/vufind/Ams.ini

The database connection is required for the following issues:

- Importing access levels

  The field Access protocol is indexed in Solr during the metadata import. During the import process, VuFind gets that value from the database to use it as a facet for filtering search results. The query is stored in this file (getAccessLevel)

  /usr/local/vufind/module/VuFind/src/VuFind/XSLT/Import/Soas.php

- Display resources

  The display of the resources at bundle level requires a database connection to show the resource list: name, access level, type, url. The action is encoded in this file (getResource):

  /usr/local/vufind/module/VuFind/src/VuFind/ILS/Driver/Ams.php

- Authentication

  The login requires a database connection against AMS. The query is saved in this file (patronLogin):

  /usr/local/vufind/module/VuFind/src/VuFind/ILS/Driver/Ams.php

7. WEBMASTER TOOLS

VuFind allows these webmaster tools:

- Sitemaps

A sitemap for every identified URL (deposits and bundles) is automatically created if this setting is activated. The configuration file is /usr/local/vufind/local/config/vufind. To generate the sitemaps, the administrator will execute this command:

cd /usr/local/vufind/util/
php sitemap.php

Once executed, check the path: /usr/local/vufind/public/sitemap/

- Robots.txt

  Once the metadata are published, the search engines will index them. We have included a robots.txt file in the path /url/local/vufind/public to disallow the indexation for certain URLs:

  Disallow: /AJAX

  Disallow: /Alphabrowse

  Disallow: /Author

  Disallow: /Browse

  Disallow: /Combined

  Disallow: /Search/Results

  Disallow: /Summon

  Disallow: /SummonRecord

  Moreover, the administrator should review the indexation by robots, such as Baidu and Yandex, in the file access.log. If they slow the access to the interface, control them in the firewall.

8. MAPPING

After the import of the metadata, VuFind creates its own list of data fields to allow searching by them. Some of the data from LAT were stored as the default data fields of VuFind (title, keyword…), but the specific data for LAT were analyzed and stored in new VuFind data fields created for this project. This mapping from the XML fields to the VuFind data fields is published on:

https://docs.google.com/spreadsheets/d/1QYSH7wzibLVIPolh6wAXczCv7PfggbsTkmAUxXUZD_I/edit#gid=836858062

The table shows:

- XML field
- VuFind-Solr data field
- Facet or not
- Search term or not
- Page

Easy name	Internal	XML field	SOLR field	Level	Where in Vufind	Facet	Index	Sorting	Notes
setSpec	yes	changeSpec(//identifier)	setSpec	Both	--				Used to check the resources from the DB.
Format	no	imdi:Corpus ó imdi:Session	format	Both	--	Format			
Id	no	Identifier	Id	Both	URL				
corpusid	yes	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Keys/imdi:Key[@Name='CorpusId']	corpusid	Both	--				
hierarchy_top_id	yes	getParentId(//imdi:METATRANSCRIPT/setSpec)	hierarchy_top_id	Both	--				
parent_id	yes	getParentId(//imdi:METATRANSCRIPT/setSpec)	parent_id	Both	--				
hierarchy_top_title	yes	getParentId(//imdi:METATRANSCRIPT/setSpec)	hierarchy_top_title	Both	--				
Is_hierarchy_id	yes	getParentId(//imdi:METATRANSCRIPT/setSpec)	Is_hierarchy_id	Both	--				
Is_hierarchy_title	yes	getParentId(//imdi:METATRANSCRIPT/setSpec)	Is_hierarchy_title	Both	--				
continent	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Location/imdi:Continent	Deposit_continent	Deposit	--				
Deposit Name	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:Name	Deposit_name	Deposit	Deposit view				
Longitude	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Keys/imdi:Key[@Name='Longitude']	deposit_longitude	Deposit	Deposit view				Used to display the map
Latitude	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Keys/imdi:Key[@Name='Latitude']	Deposit_latitude	Deposit	Deposit view				Used to display the map
Deposit Status	no	statusCode(//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Keys/imdi:Key[@Name='Deposit_Status'])	Deposit_status	Deposit	--	Status			
Deposit Status (long)	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Keys/imdi:Key[@Name='Deposit_Status']	deposit_status_string	Deposit	Deposit view				
Deposit Change Date	yes	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Keys/imdi:Key[@Name='StatusInfo_ChangeDate']	Deposit_change_date	Deposit	--				
Funding Body	no	getFundingBody(//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Project/imdi:Id)	project_id	Deposit	--	Funding body			
Creation date	yes	getFirstIndexed(//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Keys/imdi:Key[@Name='CorpusId'])	first_indexed	Deposit	--				
Modification date	yes	getLastIndexed(//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Keys/imdi:Key[@Name='CorpusId'])	last_indexed	Deposit	--			Date asc, date desc	
Title	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:Title	title	Deposit	Deposit view		Title	Title	If title is empty we use //imdi:METATRANSCRIPT/imdi:Corpus/imdi:Name
Deposit Title	yes	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:Title	deposit_title	Deposit	--				If title is empty we use //imdi:METATRANSCRIPT/imdi:Corpus/imdi:Name
Clean title	yes	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:Title	clean_title	Deposit	--				If title is empty we use //imdi:METATRANSCRIPT/imdi:Corpus/imdi:Name
deposit_description	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:Description	deposit_description	Deposit	Deposit view				
summary_of_deposit	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:Description[@Name='summary_of_deposit'])	summary_of_deposit	Deposit	Deposit view				

groups_represented	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:Description[@Name='groups_represented'])	groups_represented	Deposit	Deposit view				
deposit_contents	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:Description[@Name='deposit_contents'])	deposit_contents	Deposit	Deposit view				
short_description	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:Description[@Name='short_description'])	short_description	Deposit	Results view (deposit)				
special_characteristics	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:Description[@Name='special_characteristics'])	special_characteristics	Deposit	Deposit view				
history_of_deposit	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:Description[@Name='history_of_deposit'])	history_of_deposit	Deposit	Deposit view				
other_information	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:Description[@Name='other_information'])	other_information	Deposit	Deposit view				
acknowledgement	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:Description[@Name='acknowledgement'])	acknowledgement	Deposit	Deposit view				
bundle_language_name	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Content/imdi:Languages/imdi:Language/imdi:Name	bundle_language_name	Deposit	--	Language			
bundle_language_name_string	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Content/imdi:Languages/imdi:Language/imdi:Name	bundle_language_name_string	Deposit	Results view (deposit) and Deposit View				
deposit_cover_image	no	changeDepositLink(//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Keys/imdi:Key[@Name='CoverImage'])	deposit_cover_image	Deposit	Deposit view				
depositor	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Actors/imdi:Actor/imdi:Name	depositor	Deposit	Results view (deposit) and Deposit view and Author view				
deposit_id	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Actors/imdi:Actor/imdi:Code	deposit_id	Deposit	Deposit view				Used as depositor id for Authority module
depositor_nationality	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Actors/imdi:Actor/imdi:Keys/imdi:Key[@Name='Nationality']	depositor_nationality	Deposit	Author view. Additional Tab				
depositor_affiliation	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Actors/imdi:Actor/imdi:Keys/imdi:Key[@Name='Affiliation']	depositor_affiliation	Deposit	Deposit view				
author	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Actors/imdi:Actor/imdi:Name	author	Deposit	Author view.				
depositor_image	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Actors/imdi:Actor/imdi:Keys/imdi:Key[@Name='Picture']	depositor_image	Deposit	Deposit view. Author view				
language_information	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Content/imdi:Languages/imdi:Description	language_information	Deposit	Deposit view				
deposit_langua e_name	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Content/imdi:Languages/imdi:Language/imdi:Name	deposit_langua e_name	Deposit	Deposit view				
deposit_language_name_iso	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Content/imdi:Languages/imdi:Language/imdi:Id	deposit_language_name_iso	Deposit	Deposit view				
deposit_country_name	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Location/imdi:Country	deposit_country_name	Deposit	Deposit view				
country_name	no	//imdi:METATRANSCRIPT/imdi:Corpus/imdi:MDGroup/imdi:Location/imdi:Country	country_name	Deposit	--	Country			

resource_access_protocol	yes	getAccessLevel(//identifier)	resource_access_protocol	Bundle	--				NOT USED (Only test)
title		ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:Title	title	Bundle	Results view (bundle) and Bundle view		Title	Title	If title if emtpy we use ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:Name
bundle_title		ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:Title	bundle_title	Bundle	--				If title if emtpy we use ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:Name
clean_title		ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:Title	clean_title	Bundle	--				If title if emtpy we use ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:Name
bundle_date_created		ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:Date	bundle_date_created	Bundle	--				
bundle_location		ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:MDGroup/imdi:Location/imdi:Address	bundle_location	Bundle	Bundle view				
bundle_location_string		ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:MDGroup/imdi:Location/imdi:Address	bundle_location_string	Bundle	--				
country_name		ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:MDGroup/imdi:Location/imdi:Country	country_name	Bundle	Bundle view				
bundle_description		ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:Description	bundle_description	Bundle	Bundle view				
bundle_language_name		ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:MDGroup/imdi:Content/imdi:Languages/imdi:Language/imdi:Name	bundle_language_name	Bundle	Bundle view				
bundle_language_name_string		ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:MDGroup/imdi:Content/imdi:Languages/imdi:Language/imdi:Name	bundle_language_name_string	Bundle	--	Language	Language		
bundle_genre		ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:MDGroup/imdi:Content/imdi:Keys/imdi:Key[@Name='Genre']	bundle_genre	Bundle	Results view (bundle-keywords) and Bundle view				
bundle_genre_string		ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:MDGroup/imdi:Content/imdi:Keys/imdi:Key[@Name='Genre']	bundle_genre_string	Bundle	--	Genre			
bundle_tag		ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:MDGroup/imdi:Content/imdi:Keys/imdi:Key[@Name='Tag']	bundle_tag	Bundle	Results view (bundle-keywords) and Bundle view				
bundle_tag_string		ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:MDGroup/imdi:Content/imdi:Keys/imdi:Key[@Name='Tag']	bundle_tag_string	Bundle	--	Tags			
bundle_participant		ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:MDGroup/imdi:Actors/imdi:Actor/imdi:Name	bundle_participant	Bundle	Results view (bundle-keywords) and Bundle view	Participants			
resource_link		../resources/<xsl:value-of select="ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:MDGroup/imdi:Keys/imdi:Key[@Name='CorpusId']]	resource_link	Bundle	--				MediaFile
resource_type		ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:Resources/imdi:MediaFile/imdi:Type	resource_type	Bundle	Results view (bundle-keywords) and Bundle view				MediaFile

resource_type_string		ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:Resources/imdi:MediaFile/imdi:Type	resource_type_string	Bundle	--	Type			MediaFile
resource_keyword		ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:MDGroup/imdi:Content/imdi:Keys/imdi:Key[@Name='Topic']	resource_keyword	Bundle	Results view (bundle-keywords) and Bundle view		Keyword		
resource_keyword_string		ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:MDGroup/imdi:Content/imdi:Keys/imdi:Key[@Name='Topic']	resource_keyword_string	Bundle	--	Topic			
resource_link		../resources/<xsl:value-of select="ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:MDGroup/imdi:Keys/imdi:Key[@Name='CorpusId']]	resource_link	Bundle	--				Written Resource. Not used (test)
resource_type		ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:Resources/imdi:WrittenResource/imdi:Type	resource_type	Bundle	--	Type			Written Resource
resource_type_string		ancestor::imdi:METATRANSCRIPT/imdi:Session/imdi:Resources/imdi:WrittenResource/imdi:Type	resource_type_string	Bundle	Results view (bundle-keywords) and Bundle view				Written Resource

9. STATISTICS

VuFind is compliant with Google Analytics and Piwik to get statistics. To configure it, the administrator should modify the file /usr/local/vufind/local/config/vufind/config.ini.

Currently, Google Analytics is already configured and the account number is: UA-56492944-2

10. FAQ

Some FAQs:

- Vufind is not updated

  Most of the times, the cause is a harvesting error. The administrator should verify that the cron tasks are being executing. Logs are stored in: /usr/local/vufind/harvest.log

- VuFind doesn't harvest

  The cause is the harvest_oai.php command has not been executed. The administrator should check the file /usr/local/vufind/harvest/oai.ini to manually create a URL using the values got from host and set variables:

  https://lat1.lis.soas.ac.uk/ds/oaiprovider/oai2?verb=ListRecords&metadataPrefix=imdi&set=MPI0:MPI43292:MPI663110&from=2016-02-24

  Example:

- https://lat1.lis.soas.ac.uk/ds/oaiprovider/oai2 : host value
- MPI0:MPI43292:MPI663110: set value
- 2016-02-24: date of the file /usr/local/vufind/local/harvest/ELAR/last_harvest.txt

If that URL fails, the administrator should contact the metadata provider to ask them for data fixes.

The error might be in the metadata. The administrator should verify it checking the file /usr/local/vufind/local/harvest/ELAR/soas-lat-harvest.log. In that file there is a list of the harvested files.

- VuFind doesn't import

  If VuFinf doesn't import records, the service should be stopped to verify if the search engine (Solr) is down. To stop the service use:

  cd /usr/local/vufind/
  ./vufind.sh restart

  If the problem continues, the administrator should verify if the generated XML has any errors. The log /usr/local/vufind/solr/logs/solr.log will show that information.

- VuFind is down

  If VuFind is down, the administrator should execute this command:

  cd /usr/local/vufind/
  ./vufind.sh restart

- Changes to VuFind texts

All the English texts are stored in the file /usr/local/vufind/languages/en.ini. To edit that file, it is recommended to download first the file. Before upload again the file, the administrator should verify that the character codification is UTF-8. To see the changes, it is necessary to empty the cache: rm -Rf /usr/local/vufind/local/cache/languages/*