MUSE

MuseKnowledge™ Hybrid Search

MuseGlobal, Inc.
One Embarcadero
Suite 500
San Francisco, CA 94111
415 896-6873
www.museglobal.com

MuseGlobal S.A
Calea Bucuresti
Bl. 27B, Sc. 1, Ap. 10
Craiova, România
40 251-413496
www.museglobal.ro

EduLib, S.R.L.
Calea Bucuresti
Bl. 27B, Sc. 1, Ap. 2
Craiova, România
40 351-420970
www.edulib.com

Version: 1.2
Date: 14th November 2016
Author: EduLib, S.R.L.

Currently...

There are two basic types of Search Engines

- Indexing Search
  - A local Indexing Search creates an index from a repository of records, often on a ”just in case” basis. The repository may be local (publisher, aggregator, etc.) or may be as wide as the whole Web (Bing, Baidu, Google, etc.);
  - Proprietary web index, such as Bing, Baidu, Google, etc.;
  - Open Source Search (OSS) server, such as Apache Solr.
- Federated Search
  - A local Federated Search translates the user search and sends it to a number of remote Indexing Search Engines, and co-ordinates the results. Since there is no index this is an ad hoc, “just in time” search for each user;
  - Only proprietary systems, largely due to the ongoing maintenance cost of the Connectors to the remote Sources;
  - Muse, with its unique connector technology does this and is properly fit in the Federated Search category.

Problems and Opportunities

Both types of Search Engines have their strengths and weaknesses

- Indexing Search
  - Weaknesses
    - Not all publishers provide meta-data;
    - Lack of transparency, what is being indexed (which Journals/Databases), what period is covered;
    - Out of date records due to delayed record indexing;
    - Only metadata is indexed;
    - Large, resource consuming software systems;
    - Records are indexed and their index possibly never used;
    - Maintenance of the index is an administrative chore due to various delivery formats and types;
  - Strengths
    - Having all records in the result set;
    - Facets features and filtering;
    - Browsing features;
    - Query suggestions, spelling;

Problems and Opportunities

- Federated Search
  - Weaknesses
    - Inconsistent search results, depending on the Source;
    - Slow response times
      - because of the extra communications involved;
      - because of the need to process every result record for normalization;
    - Incomplete coverage;
    - Unable to rank results well (meta-data differences, lack of info);
    - Brings only a limited number of results from each searched source;
  - Strengths
    - The returned records are up to date, e.g. the latest information is immediately available with no efforts at all;
    - Integrate publisher platforms on various protocols: Atom, HTTP/HTML, HTTP/XML, JSON, NCIP, OAI-PMH, RSS1.0, RSS2.0, SIP2, SQL, SRU, SRW, Telnet, Z39.50;
    - The returned records match the native platforms;
    - Specialized research: medical, legal, etc.;
    - Wide range of subscribed content.

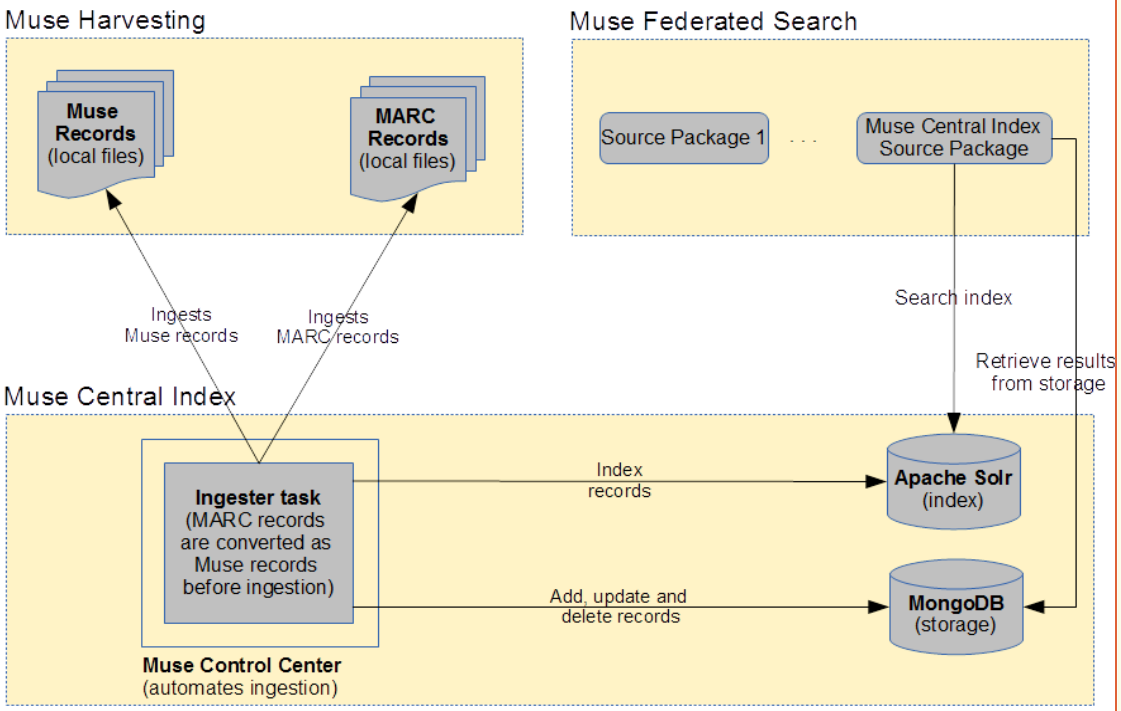
Moving Forward

Combine the strengths of Muse Federated Search platform with an available, high performance local Indexing Search to give the best of both worlds, without the drawbacks of either.

- Combine standard Indexing Search and Federated Search to optimize the powers of both;
- Run them together to give rise to MuseKnowledge™ Hybrid Search (MuseKnowledge™ HY) which provides a next generation answer to the current problems;
- Index records into Apache Solr and store the actual records content in database (MongoDB);
- Search the index and retrieve the records from the database;
- A harvesting component must exist to collect the initial set of records and incremental updates;
- Muse Control Center is the engine that drives the harvesting.

MuseKnowledge™ Hybrid Search: Components

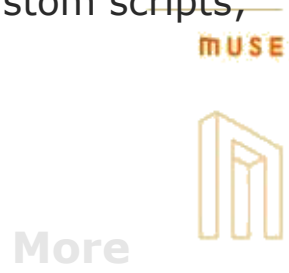
- **Muse Harvesting**
  - Harvest records from publishers; OAI-PMH and/or MARC records;
  - Initial harvesting for getting records up to current date;
  - Incremental harvesting for getting the periodical updates;
- **Muse Central Index**
  - Contains the index (Apache Solr), storage (MongoDB) and the ingester tools;
  - Records brought by the Harvesting Component are processed by the ingester tool which indexes them and stores their content in the storage;
- **Muse Federated Search Component**
  - Runs Muse Applications for end-users;
  - Source Packages that search the index and retrieve the records from the storage.



MuseKnowledge™ Harvesting

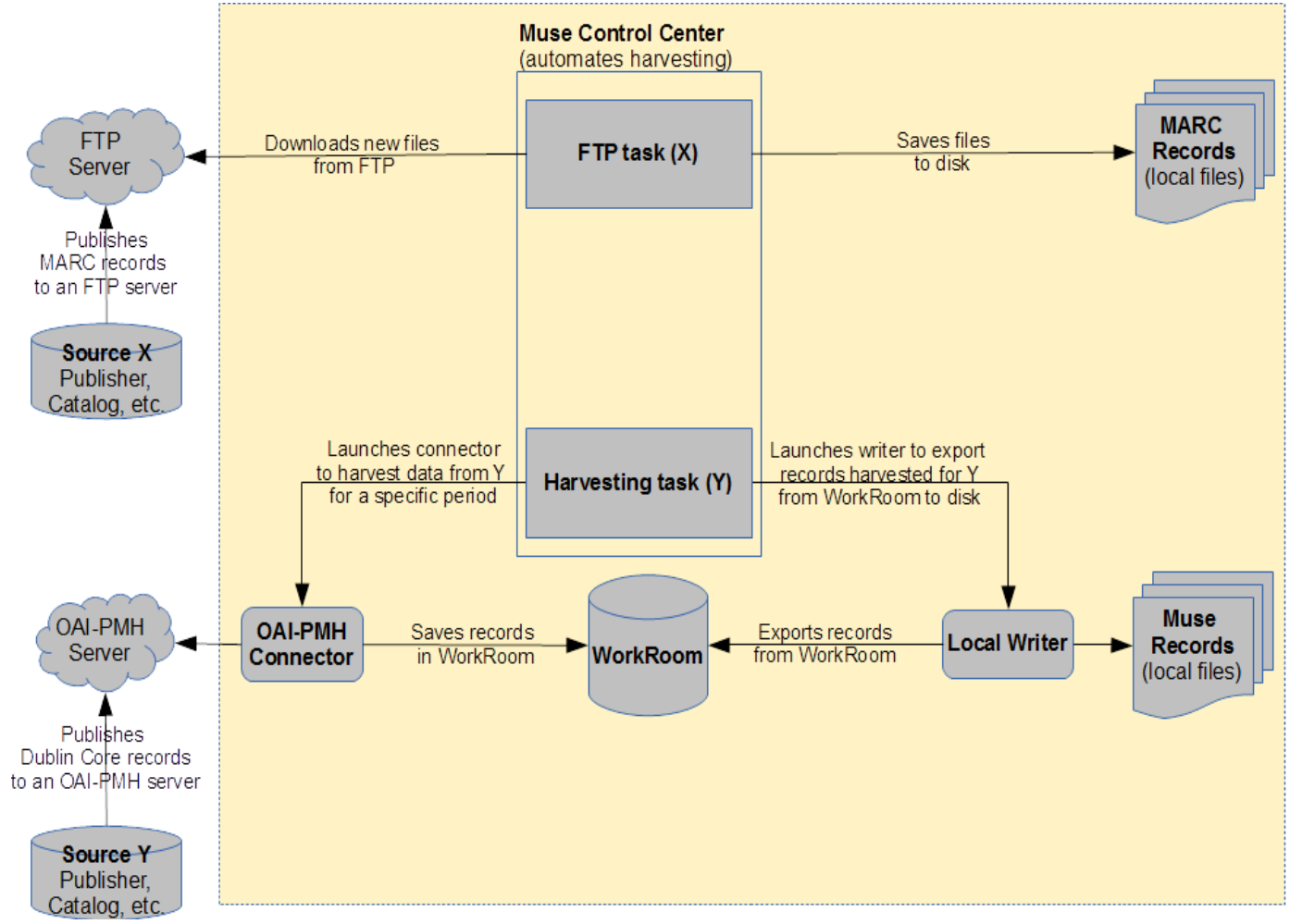
MuseKnowledge™ Harvesting is a functional system that is used to harvest records for a MuseKnowledge™ Hybrid Search System. More on "MuseKnowledge™ Harvesting Overview.pptx".

- **Harvesting Connectors**
  - Screen scraping;
  - Database (JDBC, DBF local binary files);
  - Custom XML (HTTP or local files);
- **MuseHarvesting Application**
  - Runs the Harvesting Source Packages;
  - Allows advanced editing of Muse Alerts;
- **Muse Alerts**
  - Used to run the predefined searches for harvesting;
  - Allow granular time searches on minutes, hours, days, months, years;
- **Writers**
  - Local files;
- **Muse Control Center**
  - Used for scheduling system tasks such as harvesting operations;
  - Supports other types of operations like FTP transfers, email and custom scripts; Complex workflows can be implemented via scripting.



MuseKnowledge™ Harvesting

Main MuseKnowledge™ Harvesting components in a workflow diagram



MuseKnowledge™ Harvesting: OAI-PMH

- **About OAI-PMH**
  - OAI-PMH stands for "Open Archives Initiative Protocol for Metadata Harvesting";
  - Protocol developed for harvesting (or collecting) metadata descriptions of records in an archive so that services can be built using metadata from many archives.
  - Publishers expose structured metadata via OAI-PMH;
  - It uses XML over HTTP;

- **Support in MuseKnowledge™ Harvesting**
  - A Harvesting Connector for OAI-PMH protocol is available in Muse. It can harvest data selectively (date range), in *oai_dc* metadata format;
  - OAI-PMH Source Packages exist: EmeraldOAI, NatureOAI, ArXivOAI, etc.; Generic Source Packages are also available.

MuseKnowledge™ Harvesting: MARC files

- About MARC records
  - **MA**chine-**R**eadable **C**ataloging;
  - Library catalogs keep records in MARC format;
  - Libraries provide MARC records using various methods: FTP repository, HTTP, email, etc.;
- Support in MuseKnowledge™ Harvesting
  - MuseKnowledge™ Control Center is used to download MARC files from FTP; Custom scripts can be written to download MARC from HTTP;
  - Once MARC files are obtained they are indexed;
- When indexing MARC records, the following fields are used to determine the status of a record:
  - The record id is obtained by concatenating the Control Number Identifier (003) and Control Number (001) fields;
  - The record datestamp is obtained from the Date and Time of Latest Transaction (005) field;
  - The record deleted status is obtained from the MARC record leader. If a record is marked as deleted, it will also be deleted from Muse Central Index.

MuseKnowledge™ Central Index

MuseKnowledge™ Central Index is a collection of pre-harvested metadata and full text that is searched by the MuseKnowledge™ Hybrid Service.

- The record format used is a Dublin Core-based schema;
- Can index e-book and article metadata, catalog records, and other information harvested from institutional repositories and other digital collections via the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH).
- Index
  - Apache Solr is the most popular enterprise search engine;
- Storage
  - The storage keeps all the records available in the Muse Central Index;
  - MongoDB is a popular database engine.
- Ingester
  - The process of adding or updating records in Muse Central Index is called ingesting;
  - The MuseKnowledge™ Central Index Ingester can handle MARC records and Muse records;
  - Ingested records are indexed in Apache Solr and stored in MongoDB.

About Apache Solr and MongoDB

Apache Solr

- Open source enterprise search platform built on Apache Lucene™ - java-based indexing and search technology;
- Powerful matching capabilities including phrases, wildcards, joins, grouping and much more across any data type;
- Schema driven indexing;
- Providing faceted search and filtering, query suggestions and spelling;
- Support for multi-tenant architectures;
- Performance optimizations, scalability;
- Great developer and user community.

MongoDB

- Fourth most widely mentioned database engine on the web, and the most popular for document stores;
- Free and open source cross-platform document-oriented database; NoSQL database, JSON like documents with dynamic schemas;
- Provides field, range queries, regular expression searches. Queries can return specific fields of documents and also include user-defined JavaScript functions;
- Internal index for quicker retrieval of records;
- Provides high availability with replica sets;
- High scalability using sharding.

MuseKnowledge™ Hybrid Search: Setup

- Evaluate the list of publishers from metadata availability point of view;
- Contact them for confirming and getting the metadata. Clarify the delivery of the periodical updates as well;
- Set up the harvesting:
  - For OAI-PMH delivery load the corresponding OAI-PMH Source Packages into the MuseHarvesting Application and create the Muse Alert with the needed details, such as the extraction time frame. In MuseKnowledge™ Control Center setup and configure the Muse Alerts Task to run with the desired frequency for the saved alerts;
  - For MARC records delivered via FTP, set up in Muse Control Center an FTP download task.
- Set up the ingesting:
  - In MuseKnowledge™ Control Center create and configure tasks for each harvested resource; Done via an Ant type task which calls the Muse Central Index Ingester tool with the following mandatory parameters:
    - Solr URL;
    - MongoDB URI;
    - ICE Records folder (for OAI-PMH harvested records) or MARC files location;
- Searching:
  - In a MuseSearch (or MuseKnowledge™) Application add and configure MuseCentralIndex Source Packages;
  - Generic Source Packages are also available.

Linking to Full Text

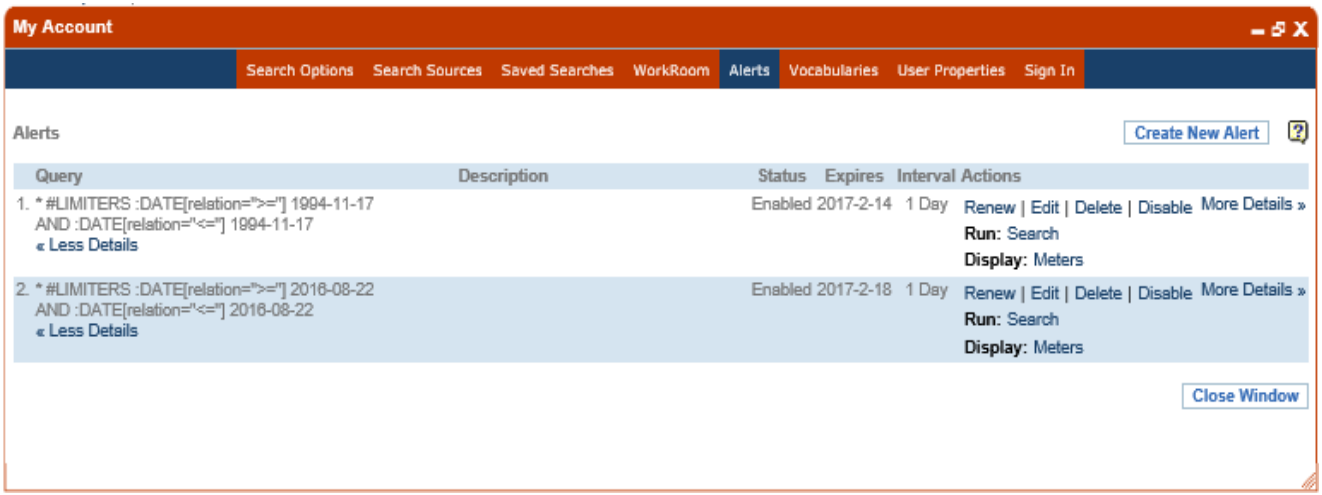
- Mandatory to link to publisher's platform for the full text;
- Usually the provided metadata contain URLs to link to the record/journal/book on the native website;
- If an URL is not available build one dynamically if possible. If a DOI identifier is available use it to form the URL to a DOI System Proxy Server, like dx.doi.org; Example:
  http://dx.doi.org/10.1109/JSEE.2013.00023
- Authentication to the publisher's platform needs to be addressed as well; This is done with MuseKnowledge™ Proxy rewriting, e.g. all record URLs are being custom rewritten by appending it to the proxy prefix URL. Example:
  http://PROXY_HOST:PROXY_PORT/ProxyApplication?qurl=RECORD_URL

- The MuseKnowledge™ Proxy Application (*ProxyApplication*) must contain source profiles that cover the rewriting of all publishers URLs. A source profile for the dx.doi.org resource must exist as well;

- The authentication mechanism to the MuseKnowledge™ Proxy Application must be considered as well: User/Password files, client IP addresses, client referer URL, standard or custom authentication methods (LDAP, IMAP, SQL, FTP), SAML, HMAC;

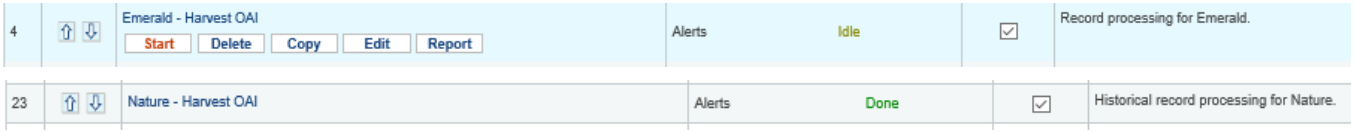
MuseKnowledge Hybrid Search Demonstration

- List of Publishers: Springer/Nature, Thomson Reuters, Elsevier, Wolters Kluwer, Emerald Insight, IET, SAGE/AMDigital, Wiley, Gale, Digital Content Associates (Zinio, Atomic Training, Rosen Digital, InfoBase, ME Books, PressReader), RSC, Britannica, DarAlMandumah, Discovery, IEEE, Cambridge University Press, Ebsco, ProQuest;
- OAI-PMH providing publishers: Nature, Emerald;
- MARC files providing publishers: Springer, Wolters Kluwer, IET, SAGE, AMDigital, Gale, RSC, IEEE, Ebsco, InfoBase;
- Harvesting:
  - Harvesting Application: MuseHarvesting, Source Packages installed: EmeraldOAI, NatureOAI;
  - Muse Alerts set:



MuseKnowledge Hybrid Search Demonstration

- Muse Control Center tasks loaded:
  - For OAI-PMH harvesting

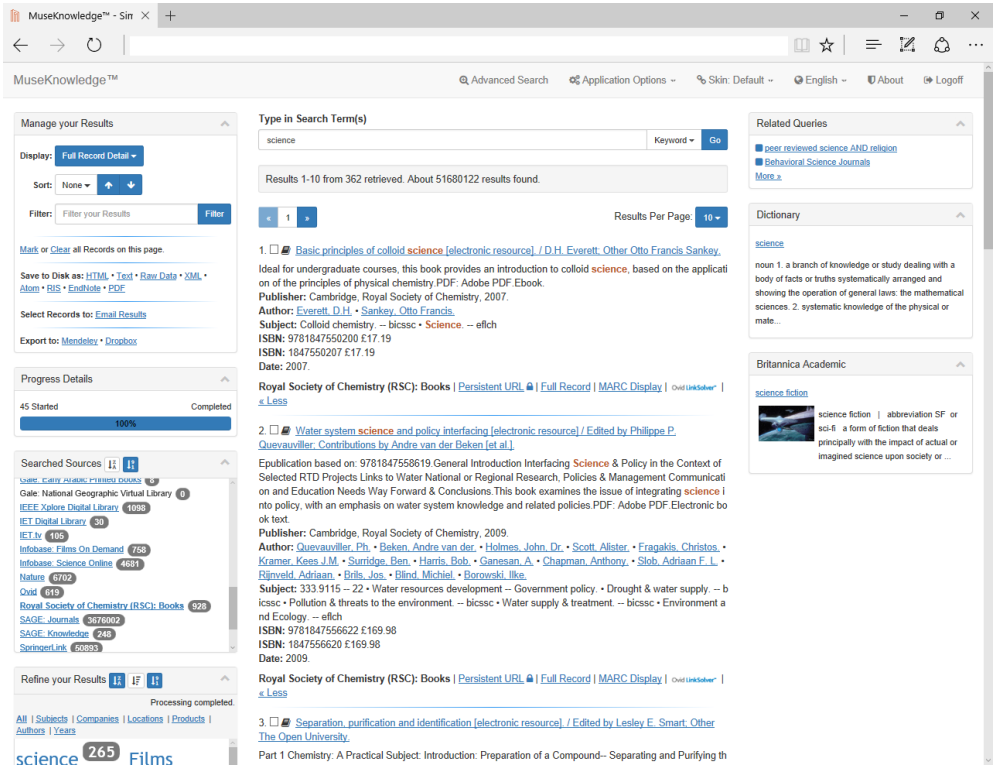


- Ingesting:
  - For OAI-PMH and MARC files;

1	Muse Control Center Scheduler	Scheduler	Running	☑	This task generates time events for the other tasks to trigger at different moments of time (hourly, daily, weekly or monthly).
2	Reset	Ant	Idle	☑	No description available.
3	SpringerProtocols - Ingest MARC	Ant	Idle	☑	Record ingesting for Springer Protocols.
4	Emerald - Harvest OAI (Start Delete Copy Edit Report)	Alerts	Idle	☑	Record processing for Emerald.
5	Emerald - Ingest XML	Ant	Idle	☑	Record ingesting for Emerald.
6	AMDigitalCPA - Ingest MARC	Ant	Idle	☑	Record ingesting for Adam Matthew: Confidential Print: Africa, 1834-1966
7	AMDigitalCPME - Ingest MARC	Ant	Idle	☑	Record ingesting for Adam Matthew: Confidential Print: Middle East, 1839-1969
8	IEEE - Ingest MARC	Ant	Idle	☑	Record ingesting for IEEE Xplore Digital Library.
9	RSCBooks - Ingest MARC	Ant	Idle	☑	Record ingesting for RSCBooks.
10	GaleMiddleEastArabIsraeli - Ingest MARC	Ant	Idle	☑	Record ingesting for Gale: the Middle East Online: Arab-Israeli relations, 1917-1970
11	GaleMiddleEastIraq - Ingest MARC	Ant	Idle	☑	Record ingesting for Gale: the Middle East Online: Iraq, 1914-1974
12	GaleNGMA - Ingest MARC	Ant	Idle	☑	Record ingesting for Gale: National Geographic Magazine Archive
13	SAGEKnowledge - Ingest MARC	Ant	Idle	☑	Record ingesting for SAGEKnowledge.
14	Ovid - Ingest MARC	Ant	Idle	☑	Record ingesting for Ovid.
15	EbscoArabWorldResearch - Ingest MARC	Ant	Idle	☑	Record ingesting for Ebsco Arab World Research.
16	EbscoPrimarySearch - Ingest MARC	Ant	Idle	☑	Record ingesting for Ebsco Primary Search.
17	EbscoMiddleSearchPlus - Ingest MARC	Ant	Idle	☑	Record ingesting for Ebsco Middle Search Plus.
18	EbscoGreenFILE - Ingest MARC	Ant	Idle	☑	Record ingesting for Ebsco GreenFILE.
19	EbscoMASUltra - Ingest MARC	Ant	Idle	☑	Record ingesting for Ebsco MAS Ultra.
20	InfobaseScienceOnline - Ingest MARC	Ant	Idle	☑	Record ingesting for Infobase Science Online.
21	SpringerEbooks - Ingest MARC	Ant	Idle	☑	Record ingesting for Springer Ebooks.
22	SpringerPalgrave - Ingest MARC	Ant	Idle	☑	Record ingesting for Springer Palgrave.
23	Nature - Harvest OAI	Alerts	Done	☑	Historical record processing for Nature.
24	Nature - Ingest XML	Ant	Done	☑	Record ingesting for Nature.
25	TheIET - Ingest MARC	Ant	Idle	☑	Record ingesting for TheIET.

MuseKnowledge Hybrid Search Demonstration

- Search:
  - MuseSearch Application with Muse Central Index Source Packages;
  - Individual Source Packages for each publisher/product: MuseCentralIndexEbscoAWR, MuseCentralIndexEmerald, MuseCentralIndexIEEE, MuseCentralIndexNature, MuseCentralIndexNature, etc. .



References

- Muse Central Index.pdf
- Muse Harvesting Overview.pptx

MUSE

MuseKnowledge Hybrid Search