# Integration with external databases (Scopus, Web of Science, etc.)



Disclaimer

DSpace-CRIS is not directly or indirectly related to any commercial bibliographic database, the available integration is provided by the DSpace-CRIS developers as-is at the best of their knowledge without any guarantee of proper behaviour with such third part services and it is not an endorsement of such services

DSpace-CRIS implement three levels of integration with external databases:

- Realtime import of bibliographic records searching the external database by identifiers or title, authors
- Periodic scanning of the external database to retrieve institutional publications to import
- Retrieve of bibliometrics value for items in DSpace(-CRIS)

## Realtime import of bibliographic records

Other than the providers already available in a standard DSpace installation such as ArXiv, PubMed, Cinii, CrossRef and generic OAI-PMH providers, DSpace-CRIS add to the Submission Lookup Step the ability to query PubMed Europe, Scopus, SciVal and/or Web of Science.

### PubMed Europe

The class org.dspace.submit.lookup.PubmedEuropeFileDataLoader is the implementation of the submission lookup interface that enable the integration with the Web of Knowledge WokSearch API.

To enable the integration it is necessary to uncomment the bean definitions in the **[dspace-installDir]/config/spring/api/bte.xml file**

```
<bean name="multipleDataLoader" class="org.dspace.submit.lookup.MultipleSubmissionLookupDataLoader" scope="
prototype">
            <property name="dataloadersMap">
                <map>
                                ...
                                <!-- <entry key="pubmedEurope" value-ref="pubmedEuropeOnlineDataLoader"/> -->
                                ...
                                <!-- <entry key="pubmedEuropeXML" value-ref="pubmedEuropeFileDataLoader"/> --
>
```

pubmedEuropeXML enables the use of the XML format specific of PubMed Europe.

The metadata mapping is defined in the bean

```
<bean name="pubmedEuropeInputMap" class="java.util.HashMap" scope="prototype">
        <constructor-arg>
                <map key-type="java.lang.String" value-type="java.lang.String">
                        <entry key="pmid" value="pubmedID" />
                        <entry key="pmcid" value="pubmedcentralID" />
                        <entry key="doi" value="doi" />
                        <entry key="ISSN" value="jissn" />
                        <entry key="EISSN" value="jeissn" />
                        <entry key="jTitle" value="journal" />
                        <entry key="startPage" value="firstpage" />
                        <entry key="endPage" value="lastpage" />
                        <entry key="title" value="title" />
                        <entry key="pubDate" value="issued" />
                        <entry key="volume" value="volume" />
                        <entry key="issue" value="issue" />
                        <entry key="language" value="language" />
                        <entry key="pubType" value="subtype" />
                        <entry key="keyword" value="keywords" />
                        <entry key="primaryMeshHeading" value="meshheadings" />
                        <entry key="secondaryMeshHeading" value="meshqualifiers" />
                        <entry key="abstractText" value="abstract" />
                        <entry key="author" value="authors" />
                        <entry key="investigator" value="investigators" />
                        <entry key="publisher" value="publisher" />
                        <entry key="series" value="seriestitle" />
                        <entry key="bookTitle" value="booktitle" />
                        <entry key="isbn" value="pisbn" />
                        <entry key="sISSN" value="sissn" />
                        <entry key="edition" value="editionnumber" />
                        <entry key="url" value="url" />
                        <entry key="uri" value="uri" />
                </map>
        </constructor-arg>
</bean>
```

## Scopus

The class org.dspace.submit.lookup.ScopusOnlineDataLoader is the implementation of the submission lookup interface that enable the integration with the Scopus API.

To enable the integration it is necessary to set the following property in the **dspace.cfg** (via build.properties)

```
submission.lookup.scopus.apikey = ${submission.lookup.scopus.apikey}
```

and uncomment the bean definition in the **[dspace-installDir]/config/spring/api/bte.xml file**

```
<bean name="multipleDataLoader" class="org.dspace.submit.lookup.MultipleSubmissionLookupDataLoader" scope="
prototype">
            <property name="dataloadersMap">
                <map>
                                    ...
                                <!-- <entry key="scopus" value-ref="scopusOnlineDataLoader"/> -->
                                ...
```

the mapping is defined in the bean

```
<bean name="scopusInputMap" class="java.util.HashMap" scope="prototype">
        <constructor-arg>
                <map key-type="java.lang.String" value-type="java.lang.String">
                        <entry key="url" value="url" />
                        <entry key="eid" value="eid" />
                        <entry key="doi" value="doi" />
                        <entry key="pmid" value="pubmedID" />
                        <entry key="title" value="title" />
                        <entry key="itemType" value="subtype" />
                        <entry key="scopusType" value="providerType" />
                        <entry key="sourceTitle" value="journal" />
                        <entry key="isbn" value="pisbn" />
                        <entry key="issn" value="jissn" />
                        <entry key="eissn" value="eissn" />
                        <entry key="issued" value="issued" />
                        <entry key="volume" value="volume" />
                        <entry key="issue" value="issue" />
                        <entry key="spage" value="firstpage" />
                        <entry key="epage" value="lastpage" />
                        <entry key="description" value="abstract" />
                        <entry key="scopusKeywords" value="keywords" />
                        <entry key="articlenumber" value="articlenumber" />
                        <entry key="authors" value="authors" />
                        <entry key="authorUrl" value="authorUrl" />
                        <entry key="authorScopusid" value="authorScopusID" />
                        <entry key="orcid" value="orcid" />
                </map>
        </constructor-arg>
</bean>
```

The scopus online data provider exposes the ORCID, Scopus ID and Author Scopus URL for each authors, when not available for one or more authors the placeholder value #NODATA# is used. This allows the use of the metadata filler functionality to create from the publication richer author profiles. It also open to future development of custom BTE processor that can lookup to existent researcher profile using these IDs other than the name

## SciVal

The class org.dspace.submit.lookup.SciValOnlineDataLoader is the implementation of the submission lookup interface that enable the integration with the SciVal API.

To enable the integration it is necessary to set the following property in the **dspace.cfg** (via build.properties)

```
submission.lookup.scivalcontent.apikey = ${submission.lookup.scivalcontent.apikey}
```

and uncomment the bean definition in the **[dspace-installDir]/config/spring/api/bte.xml file**

```
<bean name="multipleDataLoader" class="org.dspace.submit.lookup.MultipleSubmissionLookupDataLoader" scope="
prototype">
            <property name="dataloadersMap">
                <map>
                                    ...
                                <!-- <entry key="scopus" value-ref="scivalOnlineDataLoader"/> -->
                                ...
```

the mapping is defined in the bean

```
<bean name="scivalInputMap" class="java.util.HashMap" scope="prototype">
        <constructor-arg>
                <map key-type="java.lang.String" value-type="java.lang.String">
                        <entry key="eid" value="eid" />
                        <entry key="doi" value="doi" />
                        <entry key="issn" value="jissn" />
                        <entry key="eissn" value="jeissn" />
                        <entry key="isbn" value="pisbn" />
                        <entry key="journalTitle" value="journal" />
                        <entry key="title" value="title" />
                        <entry key="year" value="issued" />
                        <entry key="volume" value="volume" />
                        <entry key="issue" value="issue" />
                        <entry key="edition" value="edition" />
                        <entry key="startPage" value="firstpage" />
                        <entry key="endPage" value="lastpage" />
                        <entry key="authors" value="authors" />
                        <entry key="chairs" value="chairs" />
                        <entry key="affiliations" value="affiliations" />
                        <entry key="articleNumber" value="articleNumber" />
                        <entry key="authorsWithAffiliations" value="authorsWithAffiliations" />
                        <entry key="displayUrl" value="scopusUrl" />
                        <entry key="citationCount" value="scopusCitation" />
                        <entry key="citationUrl" value="scopusCitationUrl" />
                        <entry key="url" value="url" />
                        <entry key="classificationASJC" value="classificationASJC" />
                        <entry key="keywords" value="keywords" />
                        <entry key="language" value="language" />
                        <entry key="abstracts" value="abstract" />
                        <entry key="abstractita" value="abstractita" />
                        <entry key="abstracteng" value="abstracteng" />
                        <entry key="abstractfre" value="abstractfre" />
                        <entry key="abstractger" value="abstractger" />
                        <entry key="abstractesp" value="abstractesp" />
                        <!-- <entry key="issueDate" value="issued" /> -->
                        <entry key="medium" value="medium" />
                        <entry key="titleAlternative" value="titlealternative" />
                        <entry key="issueTitle" value="issuetitle" />
                        <entry key="conferenceName" value="conferenceName" />
                        <entry key="conferenceNumber" value="conferenceNumber" />
                        <entry key="conferencePlace" value="conferencePlace" />
                        <entry key="conferenceYear" value="conferenceYear" />
                        <entry key="conferenceSponsor" value="sponsor" />
                        <entry key="conferenceTarget" value="conferencetarget" />
                        <entry key="supplement" value="supplement" />
                        <entry key="scpId" value="scopusid" />
                        <entry key="medlineId" value="medlineid" />
                        <entry key="bookTitle" value="booktitle" />
                        <!-- <entry key="#sourceAuthor" value="" /> -->
                        <!-- <entry key="#sourceTranslator" value="" /> -->
                        <entry key="publisherName" value="publisher" />
                        <entry key="publisherPlace" value="publisherPlace" />
                        <entry key="publisherCountry" value="publisherCountry" />
                        <entry key="internationalAuthor" value="internationalauthor" />
                        <entry key="itemType" value="subtype" />
                </map>
        </constructor-arg>
</bean>
```

## Web of Knowledge

The class org.dspace.submit.lookup.WOSOnlineDataLoader is the implementation of the submission lookup interface that enable the integration with the Web of Knowledge WokSearch API.

To enable the integration it is necessary to set the following property in the **dspace.cfg** (via build.properties)

```
submission.lookup.webofknowledge.user = ${submission.lookup.webofknowledge.user}
submission.lookup.webofknowledge.password = ${submission.lookup.webofknowledge.password}
```

or, if access to the web services is granted via IP

```
submission.lookup.webofknowledge.ip.authentication = ${submission.lookup.webofknowledge.ip.authentication} #
true to enable IP authentication
```

and uncomment the bean definition in the **[dspace-installDir]/config/spring/api/bte.xml file**

```
<bean name="multipleDataLoader" class="org.dspace.submit.lookup.MultipleSubmissionLookupDataLoader" scope="
prototype">
            <property name="dataloadersMap">
                <map>
                                    ...
                                <!-- <entry key="wos" value-ref="wosOnlineDataLoader"/> -->
                                ...
```

the mapping is defined in the bean

```
<bean name="wosInputMap" class="java.util.HashMap" scope="prototype">
        <constructor-arg>
                <map key-type="java.lang.String" value-type="java.lang.String">
                        <entry key="isiId" value="isiId" />
                        <entry key="doi" value="doi" />
                        <entry key="issn" value="jissn" />
                        <entry key="journalTitle" value="journal" />
                        <entry key="title" value="title" />
                        <entry key="year" value="issued" />
                        <entry key="volume" value="volume" />
                        <entry key="issue" value="issue" />
                        <entry key="startPage" value="firstpage" />
                        <entry key="endPage" value="lastpage" />
                        <entry key="authors" value="authors" />
                        <entry key="citationCount" value="wosCitation" />
                        <entry key="keywords" value="keywords" />
                        <entry key="language" value="language" />
                        <entry key="abstracts" value="abstract" />
                        <entry key="abstractita" value="abstractita" />
                        <entry key="abstracteng" value="abstracteng" />
                        <entry key="abstractfre" value="abstractfre" />
                        <entry key="abstractger" value="abstractger" />
                        <entry key="abstractesp" value="abstractesp" />
                        <entry key="publisherName" value="publisher" />
                        <entry key="publisherPlace" value="publisherPlace" />
                        <entry key="publisherCountry" value="publisherCountry" />
                        <entry key="itemType" value="subtype" />
                        <entry key="wosType" value="providerType" />
                </map>
        </constructor-arg>
</bean>
```

## ORCID

An ORCIDOnlineDataprovider exists to import publications (Works) from an ORCID profile, see ORCID Integration

ADS

An ADS OnlineDataprovider exists to import publications from the Astrophysics Data System. A valid, free, ADS API Key is required

## Periodic scanning of the external database

The system has scripts to periodically query some external data providers for new publications, map the founds to the internal DSpace metadata and use the DBMS import to finalize the import in the repository.

For each provider a two steps procedure must be followed

1. Run the script to query the external provider, creating record in the Import boundary tables (specific of each provider)
2. Run the DBMS import script to create / update the dspace items with the new information

Currently, no special operations are performed by the retrieval scripts to guess a mapping between the publication's authors and the researcher profiles already defined in the system.

the BTE corresponding data-on-line providers are used by all the scripts to convert the internal publication representational (scopus, wos, pubmed) to the internal DSpace metadata, this mean that the mapping is defined in the **[dspace-installDir]/config/spring/bte.xml** see above

## PubMed Europe

The DSpace script to invoke is

```
./dspace dsrun org.dspace.app.cris.batch.PMCEuropeFeed -p submitter -c collectionID [-q query]  [-s start_date
(yyyy-mm-dd)] [-e end_date(yyyy-mm-dd)] [-t] [-m <metadata-for-pmid>] [-n <metadata-for-pmcid>]
```

**-p** the id or the email address of the user that will be used to create / update items

**-c** the target collection for new items

**-q** the search query for pubmed. If not specified it is retrieved from the configuration file

**-s** the start date to consider for new / updated record in pubmed. By default the script will search for changes since the previous successful execution of the script or today when executed for the first time

**-e** the end date to consider (useful in conjuction with start_date to "recover" past records

**-t** the script is executed in DRY-RUN mode, the retrieved records are just displayed

**-m** specify the metadata used to store the pmid identifier, default dc.identifier.pmid

**-n** specify the metadata used to store the pmcid identifier, default dc.identifier.pmcid

The script uses the configuration file **[dspace-installDir]/config/modules/pmceuropefeed.cfg** to get default values for some of the previous properties when not specified from the command line and additional configuration properties like the service endpoint URL

## Scopus

The DSpace script to invoke is

```
./dspace dsrun org.dspace.app.cris.batch.ScopusFeed -q query -p submitter -s start_date(yyyy-mm-dd) -e end_date
(yyyy-mm-dd) [-f] -c collectionID
```

**-p** the id or the email address of the user that will be used to create / update items

**-c** the target collection for new items to use when a specific mapping is not defined in the configuration file

**-f** will force the script to use the specified collection (-c) for all the found items ignoring the mapping defined in the configuration file

**-q** the search query for pubmed. If not specified it is retrieved from the configuration file

**-s** the start date to consider for new / updated record in scopus. By default the script will search for changes from yesterday

**-e** the end date to consider (useful in conjunction with start_date to "recover" past records)

The script uses the configuration file **[dspace-installDir]/config/modules/scopusfeed.cfg** to get default values for some of the previous properties when not specified from the command line and additional configuration properties like the service endpoint URL and the mapping between Scopus publication types and Collections

```
# Article
# scopus.type.Article.collectionid=1
# Abstract Report
# scopus.type.Abstract\ Report.collectionid=1
# Article in Press
# scopus.type.Article\ in\ Press.collectionid=1
# Book
# scopus.type.Book.collectionid=1
...
```

### Web of Knowledge

The DSpace script to invoke is

```
./dspace dsrun org.dspace.app.cris.batch.WosFeed -q query -p submitter -s start_date(yyyy-mm-dd) -e end_date
(yyyy-mm-dd) [-f] -c collectionID
```

**-p** the id or the email address of the user that will be used to create / update items

**-c** the target collection for new items to use when a specific mapping is not defined in the configuration file

**-f** will force the script to use the specified collection (-c) for all the found items ignoring the mapping defined in the configuration file

**-q** the search query for Web of Knowledge. If not specified it is retrieved from the configuration file

**-s** the start date to consider for new / updated record in web of knowledge. By default the script will search for changes from yesterday

**-e** the end date to consider (useful in conjunction with start_date to "recover" past records)

The script uses the configuration file **[dspace-installDir]/config/modules/wosfeed.cfg** to get default values for some of the previous properties when not specified from the command line and additional configuration properties like the service endpoint URL and the mapping between WoK publication types and Collections

```
# wos.type.Article.collectionid=7
# wos.type.Abstract\ of\ Published\ Item.collectionid=7
# wos.type.Art\ Exhibit\ Review.collectionid=7
# wos.type.Bibliography.collectionid=7...
```

## Retrieve of bibliometrics data (citation count)

### PubMed Central

The system is able to query PubMed Central PMC to retrieve the list of citing publications for each publication in DSpace with a pmid. The functionality rely on the use of the meatadata dc.identifier.pmid to hold the pmid. An utility script is provided to enrich items that have a DOI or a PMCID with the pmid identifier.

The script is

```
org.dspace.app.cris.metrics.pmc.script.RetrievePubMedID
```

it queries the pmc SORL core using the known identifiers (dc.identifier.doi and/or dc.identifier.pmcid) and add the resulting dc.identifier.pmid if found.

The pmc SOLR core is populated from a dump of the pmc database available for free as csv file at the following URL

ftp://ftp.ncbi.nlm.nih.gov/pub/pmc/PMC-ids.csv.gz

once downloaded and gunzipped the bash script

```
[dspace-installDir]/bin/pubmed-central-retrieve
```

loads the CSV in the SOLR core for fast querying.

The process should be performed periodically if you don't plan to collect the pmid in the submission

Once your dspace items (publications) have the dc.identifier.pmid correctly set you can use the bash script

```
[dspace-installDir]/bin/pubmed-retrieve-citation-second
```

to invoke all the DSpace script needed to retrieve the PMC citation list, store the count as metrics (pubmed) of the dspace items and build the basic derivative metric such as percentile, variation over one week / month and aggregate the value to the researcher

## Scopus

```
[dspace-installDir]/bin/scopus-retrieve
```

The bash script will execute all the dspace script needed by the functionality to

- retrieve the citation count from scopus (max 5000 publications for execution, ignoring publication with citation count new than 7 days)
- count the number of publication in scopus (with a dc.identifier.eid)
- aggregate the metrics to the Researcher level

The file **[dspace-installDir]/config/modules/cris.cfg** contains some relevant configurations

```
ametrics.elsevier.scopus.enabled = ${cris.ametrics.elsevier.scopus.enabled}
ametrics.elsevier.scopus.endpoint = ${cris.ametrics.elsevier.scopus.endpoint}
ametrics.elsevier.scopus.apikey = ${cris.ametrics.elsevier.scopus.apikey}
...
#scopus id
ametrics.identifier.eid = dc.identifier.scopus
ametrics.identifier.doi = dc.identifier.doi
```

## Web of Knowledge

```
[dspace-installDir]/bin/wos-retrieve
```

The bash script will execute all the dspace script needed by the functionality to

- retrieve the citation count from web of knowledge (max 10000 publications for execution, ignoring publication with citation count new than 7 days)
- count the number of publication in wok (with a dc.identifier.isi)
- aggregate the metrics to the Researcher level

The file **[dspace-installDir]/config/modules/cris.cfg** contains some relevant configurations

```
ametrics.thomsonreuters.wos.enabled = ${cris.ametrics.thomsonreuters.wos.enabled}
ametrics.thomsonreuters.wos.endpoint = ${cris.ametrics.thomsonreuters.wos.endpoint}
...
#wos id
ametrics.identifier.ut = dc.identifier.isi
```

By default, the system expects to be granted to use the WoK webservice by IP. If you need to authenticate with username / password you need to customize the **[dspace-installDir]/config/crosswalks/wos-header.template file**