ORCID Integration

ORCID API version compatibility

Please note that ORCID API 1.2 was turned off on August 1, 2018. To use the current ORCID API 2.x, you will need DSpace 5.9 or DSpace 6.3 or newer. Details: DS-3447.

1 Introduction
2 Use case and high level benefits
3 Enabling the ORCID authority control
4 Importing existing authors & keeping the index up to date
    4.1 Different possible use cases for Index-authority script
        4.1.1 Metadata value WITHOUT authority key in metadata
        4.1.2 Metadata that already has an authority key from an external source (NOT auto-generated by DSpace)
        4.1.3 Metadata that has already a new dspace generated uid authority key
        4.1.4 Processing on records in the authority cache
    4.2 Submission of new DSpace items - Author lookup
    4.3 Admin Edit Item
    4.4 Editing existing items using Batch CSV Editing
    4.5 Storage of related metadata
5 Configuration
6 Adding additional fields under ORCID
7 Integration with other systems beside ORCID
8 FAQ
    8.1 Which information from ORCID is currently indexed in the authority cache?
    8.2 How can I index additional fields in the authority cache?
    8.3 How can I use the information stored in the authority cache?
    8.4 How to add additional metadata fields in the authority cache that are not related to ORCID?
    8.5 What happens to data if another authority control was already present?
    8.6 Where can I find the URL that is used to lookup ORCIDs?

Introduction

The ORCID integration adds ORCID compatibility to the existing solutions for Authority control in DSpace. String names of authors are still being stored in DSpace metadata. The authority key field is leveraged to store a uniquely generated internal ID that links the author to more extended metadata, including the ORCID ID and alternative author names.

This extended metadata is stored and managed in a dedicated SOLR index, the DSpace authority cache.

Use case and high level benefits

The vision behind this project consists of the following two aspects:

Lowering the threshold to adopt ORCID for the members of the DSpace community

ORCID's API has enabled developers across the globe to build points of integration between ORCID and third party applications. Up until today, this meant that members of the DSpace community were still required to implement front-end and back-end modifications to the DSpace source code in order to leverage these APIs. As DSpace aims to provide turnkey Institutional Repository functionality, the platform is expected to provide more functionality out of the box. Only an elite selection of members in the DSpace community has software development resources readily available to implement this kind of functionality. By contributing a solution directly to the core DSpace codebase, this threshold to adopt ORCID functionality in DSpace repositories is effectively lowered. The ultimate goal is to allow easy adoption of ORCID without customization of the DSpace software, by allowing repository administrators to enable or disable functionality by means of user friendly configuration.

Address generic use cases with appealing end user functionality

This proposal aims to provide user friendly features for both repository administrators as well as non- technical end users of the system. The addition of ORCID functionality to DSpace should not come at the cost of making the system more difficult for administrators and end users to use. Scope With this vision in mind, the project partners wanted to tackle the first phases for repository managers of existing DSpace repositories: ensuring that ORCIDs are properly associated with new works entering the system, as well as providing functionality to efficiently batch-update content already existing in the system, with unambiguous author identity information.

Enabling the ORCID authority control

UI Support

In DSpace 5.0 the functionality only includes user interface functionality for the XMLUI Mirage and Mirage 2 themes. Older XMLUI themes including Kubrick, Reference and Classic are currently unsupported. JSPUI is unsupported in 5.0 as well.

If you wish to enable this feature, some changes are required to the `dspace.cfg` file. The first step is to activate the authority as a valid option for authority control, this is done by adding/setting an additional plugin in the `plugin.named.org.dspace.content.authority.ChoiceAuthority` property. An example of this can be found below.

```
plugin.named.org.dspace.content.authority.ChoiceAuthority = \
    org.dspace.content.authority.SolrAuthority = SolrAuthorAuthority
```

The feature relies on the following configuration parameters in `dspace.cfg`. To activate the default settings it suffices to remove the comment hashes ("#") for the following lines. See the section at the bottom of this page what these parameters mean exactly and how you can tweak the configuration.

```
solr.authority.server=${solr.server}/authority
choices.plugin.dc.contributor.author = SolrAuthorAuthority
choices.presentation.dc.contributor.author = authorLookup
authority.controlled.dc.contributor.author = true
authority.author.indexer.field.1=dc.contributor.author
```

The final part of configuration is to add the authority consumer in front of the list of event consumers. Add "authority" in front of the list as displayed below.

```
event.dispatcher.default.consumers = authority, versioning, discovery, eperson, harvester
```

Importing existing authors & keeping the index up to date

When first enabled the authority index will be empty, to populate the authority index run the following script:

```
[dspace]/bin/dspace index-authority
```

This will iterate over every metadata under authority control and create records of them in the authority index. The metadata without an authority key will each be updated with an auto generated authority key. These will not be matched in any way with other existing records. The metadata with an authority key that does not already exist in the index will be indexed with those authority keys. The metadata with an authority key that already exist in the index will be re-indexed the same way. These records remain unchanged.

Different possible use cases for Index-authority script

Metadata value WITHOUT authority key in metadata

“Luyten, Bram” is present in the metadata without any authority key.
GOAL: “Luyten, Bram” gets added in the cache ONCE

All occurences of “Luyten, Bram” in the DSpace item metadata will become linked with the same generated uid.

Metadata that already has an authority key from an external source (NOT auto-generated by DSpace)

“Snyers, Antoine” is present with authority key “u12345”

The old authority key needs to be preserved in the item metadata and duplicated in the cache.
“u12345” will be copied to the authority cache and used as the authority key there.

Metadata that has already a new dspace generated uid authority key

Item metadata already contains an author with name “Haak, Danielle” and a uid in the authority field 3dda2571-6be8-4102-a47b-5748531ae286

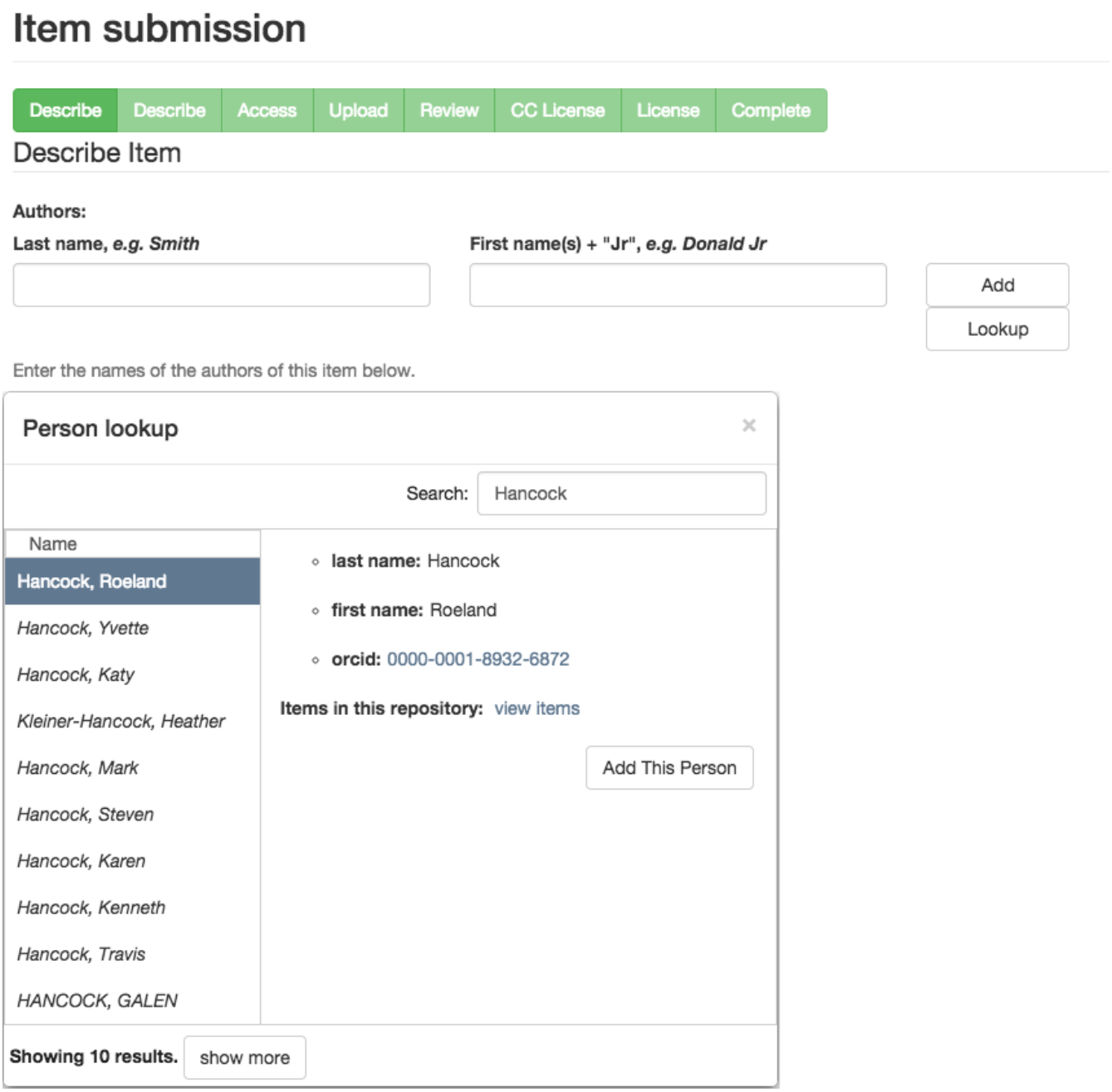
This uid is preserved and no new record is being created in the authority index.

Processing on records in the authority cache

Running this script again will update the index and keep the index clean. For example if an author occurs in a single item and that item is deleted the script will need to be run again to remove it from the index. When run again it will remove all records that no longer have a link to existing authors in the database.

Submission of new DSpace items - Author lookup

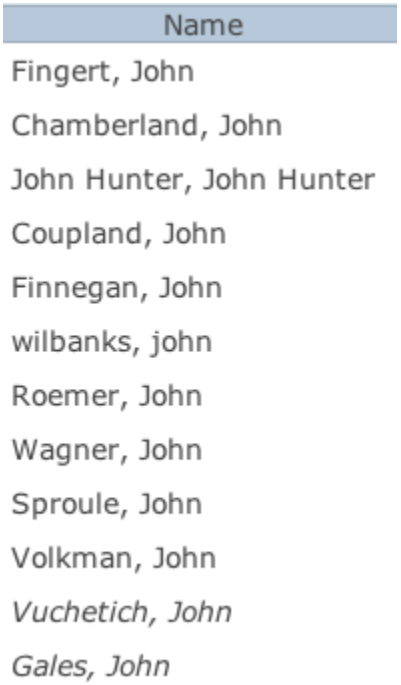
The submissions forms have not changed much. The only thing you can notice is an extra button next to the input fields for the author names. Next to the Add button, which is common for all repeatable fields, there is the Lookup & Add button.



It's by clicking on that button that the Look-up User Interface appears. If an author name was filled in but not added yet, the Lookup User Interface will immediately perform a search for that name. Otherwise the search field remains empty and a list of known authors is displayed. The list of authors is updated as you type in the search box.

Authors that already appear somewhere in the repository are differentiated from the authors that have been retrieved from ORCID.

The authors retrieved from ORCID have their name italicized and they're listed after the authors that are found in the repository.



Click on one of these names to see more information about them. The message "There's no one selected" will vanish, making room for the author's information. The available information can vary: Authors imported from ORCID have an orcid where the others do not. Authors that have been added without look-up only show their last name and first name.

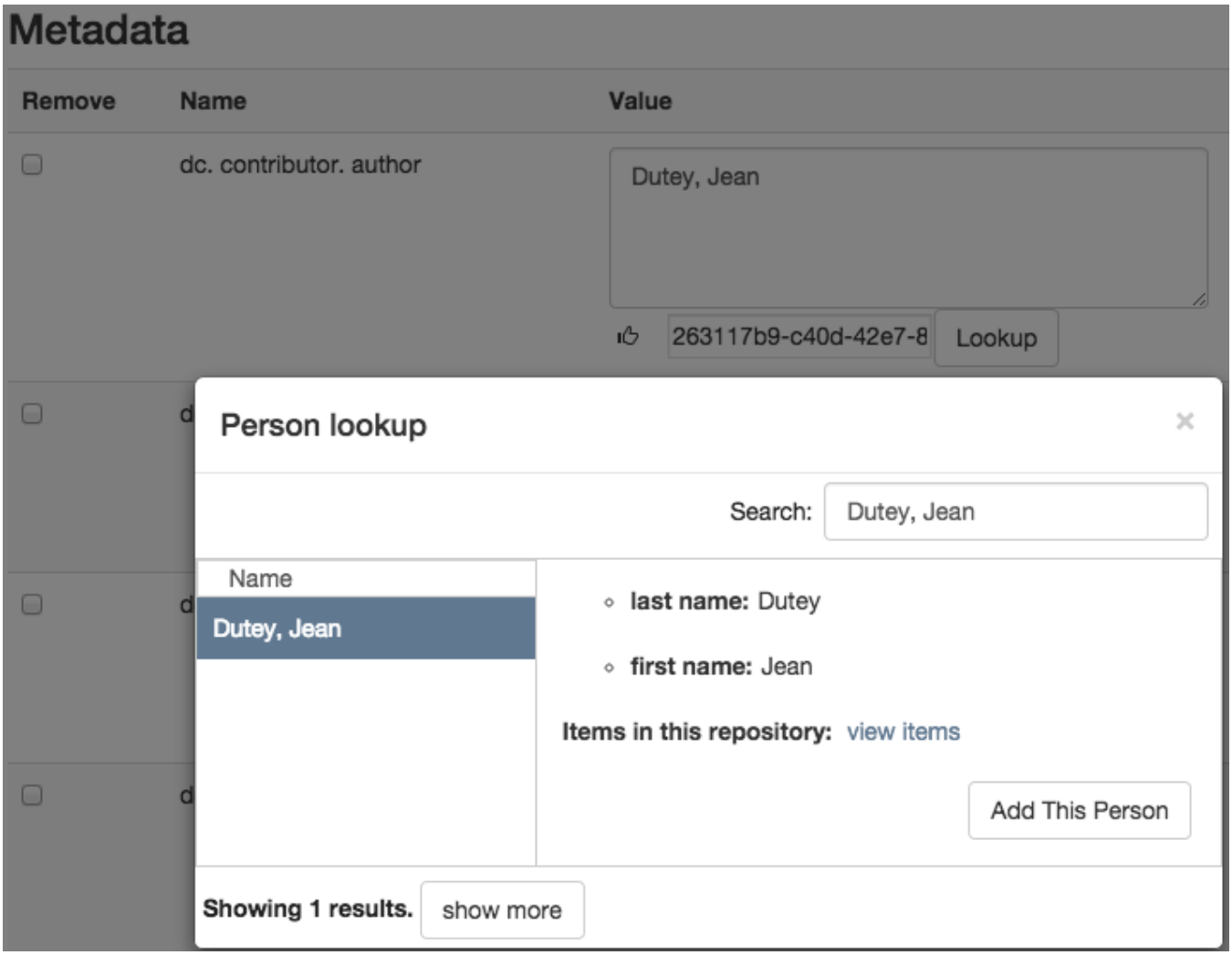
To add an author from the Look-up User Interface, you select the author in the list and then you click on the "Add This Person" button.

To add an author without look-up, you don't go through the Look-up User Interface. Instead you simply use the "Add" button in the submissions forms.

Admin Edit Item

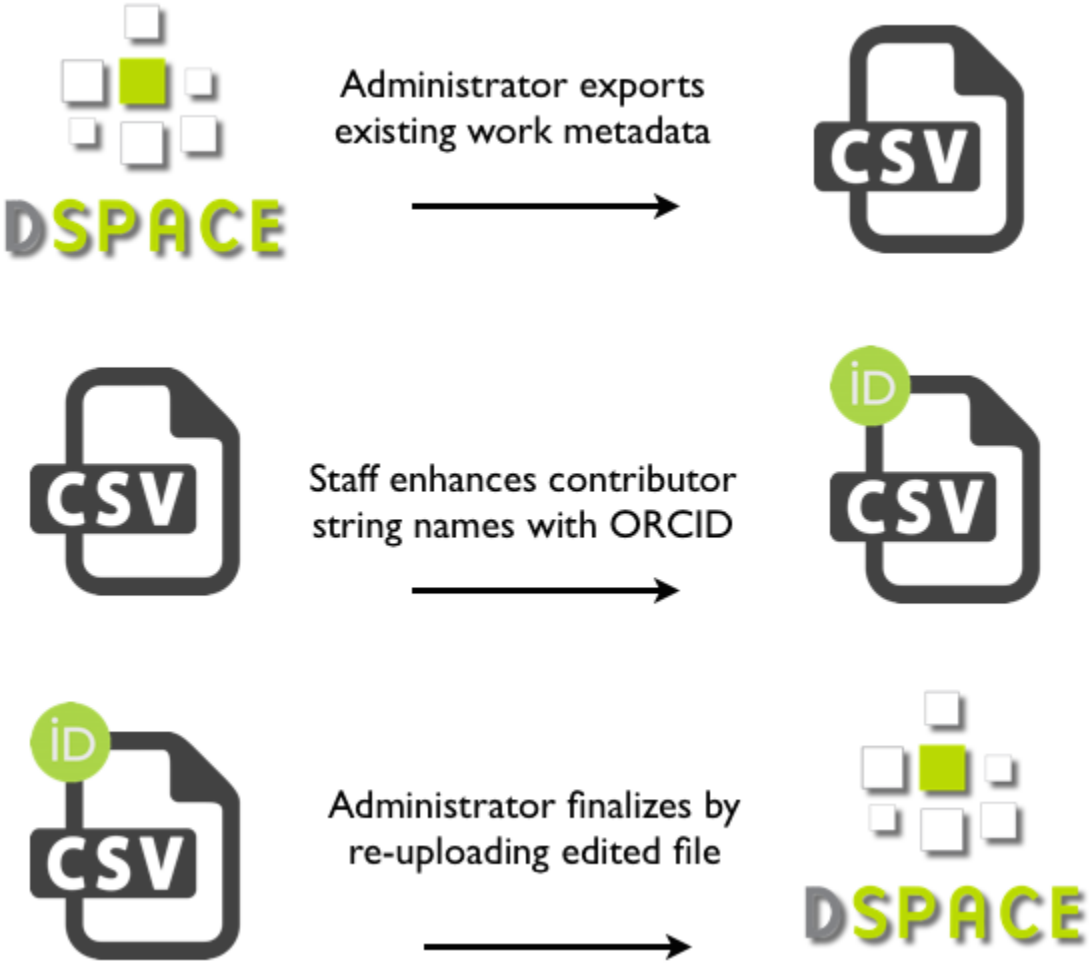
In the edit metadata page, under the values for the dc.contributor.author fields, an extra line shows the author ID together with a lock icon and a Lookup button. The author ID cannot be changed manually. However the Lookup button will help you change the author name and ID at the same time.

Clicking the Lookup button brings back the Lookup User Interface. This works just the same way as in the submission forms.



Editing existing items using Batch CSV Editing

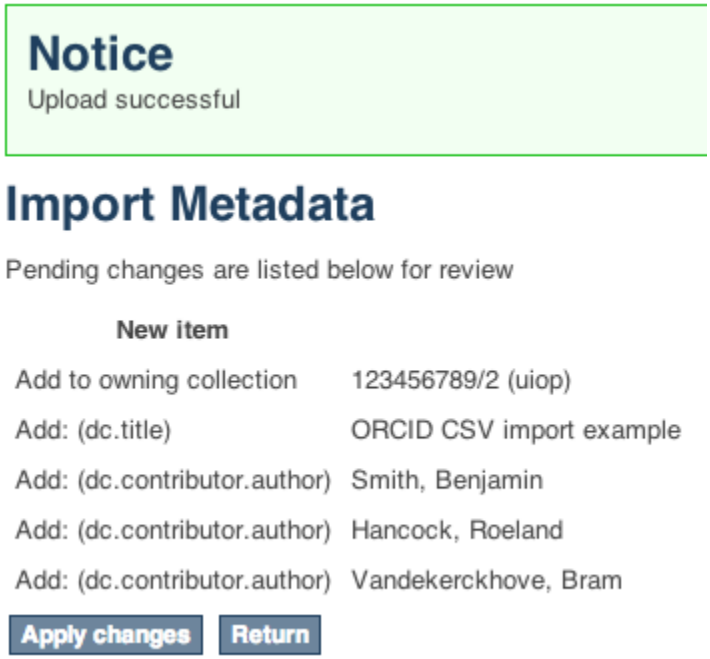
Instructions on how to use the Batch CSV Editing are found on the Batch Metadata Editing documentation page.



ORCID Integration is provided through the Batch CSV Editing feature with an extra available headers "ORCID:dc.contributor.author". The usual CSV headers only contain the metadata fields: e.g. "dc.contributor.author". In addition to the traditional header, another dc.contributor.author header can be added with the "ORCID:" prefix. The values in this column are supposed to be ORCIDs.

id	collection	dc.title	ORCID:dc.contributor.author	dc.contributor.author
+	123456789/2	ORCID CSV import example	0000-0001-8932-6872||0000-0001-9152-849X	Smith, Benjamin

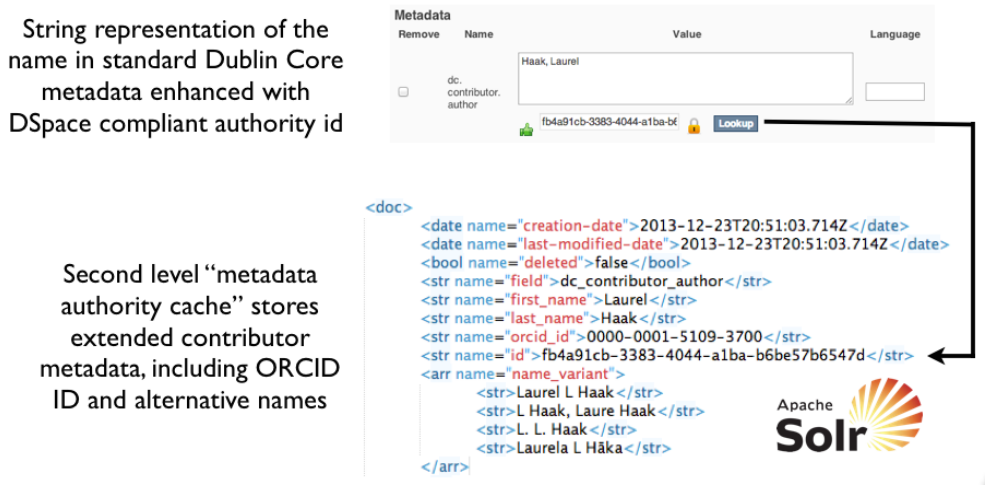
For each of the ORCID authors a lookup will be done and their names will be added to the metadata. All the non-ORCID authors will be added as well. The authority keys and solr records are added when the reported changes are applied.



Storage of related metadata

ORCID authorities not only link a digital identifier to a name. It regroups a load of metadata going from alternative names and email addresses to keywords about their works and much more. The metadata is obtained by querying the ORCID web services. In order to avoid querying the ORCID web services every time, all these related metadata is gathered in a "metadata authority cache" that DSpace can access directly.

In practice the cache is provided by an apache solr server. When a look-up is made and an author is chosen that is not yet in the cache, a record is created from an ORCID profile and added to the cache with the list of related metadata. The value of the Dublin Core metadata is based on the first and last name as they are set in the ORCID profile. The authority key for this value links directly to the solr document's id. DSpace does not provide a way to edit these records manually.



The information in the authority cache can be updated by running the following command line operation:

Command used:	`[dspace]/bin/dspace dsrun org.dspace.authority.UpdateAuthorities`
Arguments	description
-i	update specific solr records with the given internal ids (comma-separated)
-h	prints this help message

This will iterate over every solr record currently in use (unless the -i argument is provided), query the ORCID web service for the latest data and update the information in the cache. If configured, the script will also update the metadata of the items in the repository where applicable.

The configuration property can be set in `config/modules/solrauthority.cfg`:

```
auto-update-items = false | true
```

When set to true and this is script is run, if an authority record's information is updated the whole repository will be scanned for this authority. Every metadata field with this authority key will be updated with the value of the updated authority record.

Configuration

In the Enabling the ORCID authority control section, you have been told to add this block of configuration.

```
solr.authority.server=${solr.server}/authority
choices.plugin.dc.contributor.author = SolrAuthorAuthority
choices.presentation.dc.contributor.author = authorLookup
authority.controlled.dc.contributor.author = true
authority.author.indexer.field.1=dc.contributor.author
```

The ORCID Integration feature is an extension on the authority control in DSpace. Most of these properties are extensively explained on the Authority Control of Metadata Values documentation page. These will be revisited but first we cover the properties that have been newly added.

- The **`solr.authority.server`** is the url to the solr core. Usually this would be on the **`solr.server`** next to the oai, search and statistics cores.
- **`authority.author.indexer.field.1`** and the subsequent increments configure which fields will be indexed in the authority cache. However before adding extra fields into the solr cache, please read the section about Adding additional fields under ORCID.

That's it for the novelties. Moving on to the generic authority control properties:

- With the **`authority.controlled`** property every metadata field that needs to be authority controlled is configured. This involves every type of authority control, not only the fields for ORCID integration.
- The **`choices.plugin`** should be configured for each metadata field under authority control. Setting the value on SolrAuthorAuthority tells DSpace to use the solr authority cache for this metadatafield, cfr. Storage of related metadata.
- The **`choices.presention`** should be configured for each metadata field as well. The traditional values for this property are `select|suggest|lookup`. A new value, `authorLookup`, has been added to be used in combination with the SolrAuthorAuthority choices plugin. While the other values can still be used, the authorLookup provides a richer user interface in the form of a popup on the submission page.
- The browse indexes need to point to the new authority-controlled index: `webui.browse.index.2 = author:metadata:dc.contributor.*,dc.creator:text` should become **webui.browse.index.2 = author:metadataAuthority:dc.contributor.author:authority**
- More existing configuration properties are available but their values are independent of this feature and their default values are usually fine: **`choices.closed`** , **`authority.required, authority.minconfidence`** .

For the cache update script, one property can be set in `config/modules/solrauthority.cfg`:

```
auto-update-items = false | true
```

The default value for when the property is missing is false.

The final part of configuration is to add the authority consumer in front of the list of event consumers. Add "authority" in front of the list as displayed below.

```
event.dispatcher.default.consumers = authority, versioning, discovery, eperson, harvester
```

Without the consumer there is no automatic indexing of the authority cache and the metadata will not even have authority keys.

Changes to the configuration always require a server restart before they're in effect.

Adding additional fields under ORCID

Other metadata fields besides "dc.contributor.author" can benefit from the ORCID authority control at the same time. Here is an example of how to get the same ORCID functionality for the "dc.contributor.editor" metadata field assuming that "dc.contributor.author" is already configured correctly. It can be achieved by modifying configuration files only.

First add the same configuration fields that have been added for the "dc.contributor.author"

```
choices.plugin.dc.contributor.editor = SolrAuthorAuthority
choices.presentation.dc.contributor.editor = authorLookup
authority.controlled.dc.contributor.editor = true

authority.author.indexer.field.1=dc.contributor.author
authority.author.indexer.field.2=dc.contributor.editor
```

This is enough to get the look-up interface on the submission page and on the edit metadata page. The authority keys will be added and indexed with the information from orcid just as it happens with the Authors.

But you're not completely done yet, There is one more configuration step. Because now when adding new editors in the metadata that are not retrieved through the external look-up, their first and last name will not be displayed in the look-up interface next time you look for them.

To fix this, open the file at `config/spring/api/orcid-authority-services.xml` and find this spring bean:

```
<bean name="AuthorityTypes" class="org.dspace.authority.AuthorityTypes">
    <property name="types">
        <list>
            <bean class="org.dspace.authority.orcid.OrcidAuthorityValue"/>
            <bean class="org.dspace.authority.PersonAuthorityValue"/>
        </list>
    </property>
    <property name="fieldDefaults">
        <map>
            <entry key="dc_contributor_author">
                <bean class="org.dspace.authority.PersonAuthorityValue"/>
            </entry>
        </map>
    </property>
</bean>
```

The map inside the "fieldDefaults" property needs an additional entry for the editor field:

```
<entry key="dc_contributor_editor">
    <bean class="org.dspace.authority.PersonAuthorityValue"/>
</entry>
```

With this last change everything is set up to work correctly. The rest of this configuration file is meant for JAVA developers that wish to provide integration with other systems beside ORCID. Developers that wish to display other fields than first and last name can also have a look in that section.

Note: Each metadata field has a separate set of authority records. Authority keys are not shared between different metadata fields. E. g. multiple dc.contributor.author can have the same authority key and point to the same authority record in the cache. But when an ORCID is chosen for a dc.contributor.editor field, a separate record is made in the cache. Both records are updated from the same source and will contain the same information. The difference is that when performing a look-up of a person that has been introduced as an authority for an author field but not yet as an editor, it will show as record that is not yet present in the repository cache.

Integration with other systems beside ORCID

The authority cache and look-up functionality can be extended to use other sources than ORCID or to show more information in the look-up interface. However some JAVA development is necessary for this. Specific instructions can be found in the readme file of the org.dspace.authority package.

FAQ

Which information from ORCID is currently indexed in the authority cache?

Here is a breakdown of the fields stored in the solr cache.

The system/dspace related fields are: *id, field, value, deleted, creation_date, last_modified_date, authority_type*.

The fields with data coming directly from ORCID are: *first_name, last_name, name_variant, orcid_id, label_researcher_url, label_keyword, label_external_identifier, label_biography, label_country.* The field *all_labels* contains all the values from the other fields starting with "label_".

How can I index additional fields in the authority cache?

There is currently no configuration to control which fields are indexed. The only way to achieve this is to modify the source code.

List of the files at work for this job:
`config/spring/api/orcid-authority-services.xml`: OrcidSource contains the URL for orcid's REST API.
org.dspace.authority.orcid.Orcid makes the REST call
+ org.dspace.authority.orcid.xml.XMLtoBio converts the received XML to a java object (Bio).
+ org.dspace.authority.orcid.model.Bio
+ org.dspace.authority.orcid.OrcidAuthorityValue#create(org.dspace.authority.orcid.model.Bio) inserts all the values from Bio into the AuthorityValue subclass.
+ org.dspace.authority.orcid.OrcidAuthorityValue#getSolrInputDocument defines what's included in solr.

The files preceded with a '+' would be necessary to modify to add more info into the cache.

How can I use the information stored in the authority cache?

The look-up UI is currently the only place this information is sent to. However just a limited number of fields are sent. The place in the source code to modify to get more fields there is org.dspace.authority.orcid.OrcidAuthorityValue#choiceSelectMap. This is also documented in the readme of the org.dspace.authority package.

How to add additional metadata fields in the authority cache that are not related to ORCID?

Make the same configuration step as for adding additional fields under ORCID. Currently the ORCID suggestions cannot be turned off for specific fields, that would require custom code.

What happens to data if another authority control was already present?

As long as the metadata does not get indexed, there will be no changes. However every time any metadata of an item is modified, the metadata under authority control for that item will be re-indexed. When that happens a record will be inserted in the solr cache. That record's ID will be the authority key of the metadata. This can be done for all metadata at once with the index-authority script.

In short: authority keys that exist prior to enabling the solrauthority are kept. They just won't show in the look-up until they are indexed.

Where can I find the URL that is used to lookup ORCIDs?

It is found in the `config/spring/api/orcid-authority-services.xml` configuration file. Look for the `<bean name="OrcidSource">`, which is initialized with a URL of http://pub.orcid.org