DOI Digital Object Identifier

- Persistent Identifier
- DOI Registration Agencies
  - Configure DSpace to use the DataCite API
    - dspace.cfg
    - Metadata conversion
    - Identifier Service
    - DOIs using DataCite and Item Level Versioning
    - Command Line Interface
    - 'cron' job for asynchronous reservation/registration
    - Limitations of DataCite DOI support
  - Configure DSpace to use EZID service for registration of DOIs
    - Limitations of EZID DOI support
- JSPUI specific configurations
- Adding support for other Registration Agencies

Persistent Identifier

It is good practice to use Persistent Identifiers to address items in a digital repository. There are many different systems for Persistent Identifiers: Handle , DOI , urn:nbn, purl and many more. It is far out of the scope of this document to discuss the differences of all these systems. For several reasons the Handle System is deeply integrated in DSpace, and DSpace makes intensive use of it. With DSpace 3.0 the Identifier Service was introduced that makes it possible to also use external identifier services within DSpace.

DOIs are Persistent Identifiers like Handles are, but as many big publishing companies use DOIs they are quite well-known to scientists. Some journals ask for DOIs to link supplemental material whenever an article is submitted. Beginning with DSpace 4.0 it is possible to use DOIs in parallel to the Handle System within DSpace. By "using DOIs" we mean automatic generation, reservation and registration of DOIs for every item that enters the repository. These newly registered DOIs will not be used as a means to build URLs to DSpace items. Items will still rely on handle assignment for the item urls.

DOI Registration Agencies

To register a DOI one has to enter into a contract with a DOI registration agency which is a member of the International DOI Foundation. Several such agencies exist. Different DOI registration agencies have different policies. Some of them offer DOI registration especially or only for academic institutions, others only for publishing companies. Most of the registration agencies charge fees for registering DOIs, and all of them have different rules describing for what kind of item a DOI can be registered. To make it quite clear: to register DOIs with DSpace you have to enter into a contract with a DOI registration agency.

DataCite is an international initiative to promote science and research, and a member of the International DOI Foundation. The members of DataCite act as registration agencies for DOIs. Some DataCite members provide their own APIs to reserve and register DOIs; others let their clients use the DataCite API directly. Starting with version 4.0 DSpace supports the administration of DOIs by using the DataCite API directly or by using the API from EZID (which is a service of the University of California Digital Library). This means you can administer DOIs with DSpace if your registration agency allows you to use the DataCite API directly or if your registration agency is EZID.

Configure DSpace to use the DataCite API

If you use a DOI registration agency that lets you use the DataCite API directly, you can follow the instructions below to configure DSpace. In case EZID is your registration agency the configuration of DSpace is documented here: Configure DSpace to use EZID service for registration of DOIs.

To use DOIs within DSpace you have to configure several parts of DSpace:

- enter your DOI prefix and the credentials to use the API from DataCite in dspace.cfg,
- configure the script which generates some metadata,
- activate the DOI mechanism within DSpace,
- configure a cron job which transmits the information about new and changed DOIs to the registration agency.

dspace.cfg

After you enter into a contract with a DOI registration agency, they'll provide you with user credentials and a DOI prefix. You have to enter these in the dspace cfg. Here is a list of DOI configuration options in dspace.cfg:

Configuration File:	**[dspace]/config/dspace.cfg**
Property:	`identifier.doi.user`
Example Value:	`identifier.doi.user = user123`

Informational Note:	Username to login into the API of the DOI registration agency. You'll get it from your DOI registration agency.
Property:	`identifier.doi.password`
Example Value:	`identifier.doi.password = top-secret`
Informational Note:	Password to login into the API of the DOI registration agency. You'll get it from your DOI registration agency.
Property:	`identifier.doi.prefix`
Example Value:	`identifier.doi.prefix = 10.5072`
Informational Note:	The prefix you got from the DOI registration agency. All your DOIs start with this prefix, followed by a slash and a suffix generated from DSpace. The prefix can be compared with a namespace within the DOI system.
Property:	`identifier.doi.namespaceseparator`
Example Value:	`identifier.doi.namespaceseparator = dspace-`
Informational Note:	This property is optional. If you want to use the same DOI prefix in several DSpace installations or with other tools that generate and register DOIs it is necessary to use a namespace separator. All the DOIs that DSpace generates will start with the DOI prefix, followed by a slash, the namespace separator and some number generated by DSpace. For example, if your prefix is 10.5072 and you want all DOIs generated by DSpace to look like 10.5072/dspace-1023 you have to set this as in the example value above.
Property:	crosswalk.dissemination.DataCite.publisher
Example Value:	crosswalk.dissemination.DataCite.publisher = My University Press
Informational Note:	The name of the entity which published the item.
Property:	crosswalk.dissemination.DataCite.hostingInstitution
Example Value:	crosswalk.dissemination.DataCite.hostingInstitution = My University
Informational Note:	The name of the entity which hosts this instance of the object. If not configured, it will default to the value of crosswalk. dissemination.DataCite.publisher.
Property:	crosswalk.dissemination.DataCite.dataManager
Example Value:	crosswalk.dissemination.DataCite.dataManager = My University Department of Geology
Informational Note:	If not configured, it will default to the value of crosswalk.dissemination.DataCite.publisher.

Please don't use the test prefix 10.5072 with DSpace. The test prefix 10.5072 differs from other prefixes: It answers GET requests for all DOIs even for DOIs that are unregistered. DSpace checks that it mint only unused DOIs and will create an Error: "Register DOI ... failed: DOI_ALREADY_EXISTS". Your registration agency can provide you an individual test prefix, that you can use for tests.

Metadata conversion

To reserve or register a DOI, DataCite requires that metadata be supplied which describe the object that the DOI addresses. The file [dspace]/config /crosswalks/DIM2DataCite.xsl controls the conversion of metadata from the DSpace internal format into the DataCite format. You have to add your DOI prefix, namespace separator and the name of your institution to this file, on lines 16 and 18 respectively in the following excerpt:

[dspace]/config/crosswalks/DIM2DataCite.xsl

```
<!--
    Document   : DIM2DataCite.xsl
    Created on : January 23, 2013, 1:26 PM
    Author     : pbecker, ffuerste
    Description: Converts metadata from DSpace Intermediat Format (DIM) into
                 metadata following the DataCite Schema for the Publication and
                 Citation of Research Data, Version 2.2
-->
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                xmlns:dspace="http://www.dspace.org/xmlns/dspace/dim"
                xmlns="http://datacite.org/schema/kernel-2.2"
                version="1.0">

    <!-- CONFIGURATION -->
    <!-- Please add your DOI-Prefix and your namespace separator here (e.g. 10.5072-dspace-). -->
    <xsl:variable name="prefix">10.5072-dspace-</xsl:variable>
    <!-- The content of the following variable will be used as element publisher. -->
    <xsl:variable name="publisher">My University</xsl:variable>
    <!-- The content of the following variable will be used as element contributor with contributorType
datamanager. -->
    <xsl:variable name="datamanager"><xsl:value-of select="$publisher" /></xsl:variable>
    <!-- The content of the following variable will be used as element contributor with contributorType
hostingInstitution. -->
    <xsl:variable name="hostinginstitution"><xsl:value-of select="$publisher" /></xsl:variable>
    <!-- Please take a look into the DataCite schema documentation if you want to know how to use these
elements.
         http://schema.datacite.org -->


    <!-- DO NOT CHANGE ANYTHING BELOW THIS LINE EXCEPT YOU REALLY KNOW WHAT YOU ARE DOING! -->
...
```

Just change the value in the variable named "publisher".

If you want to know more about the DataCite Schema, have a look at the documentation. If you change this file in a way that is not compatible with the DataCite schema, you won't be able to reserve and register DOIs anymore. Do not change anything if you're not sure what you're doing.

You can test the functionality of your DataCite crosswalk by running the following command in a shell:

```
[dspace]/bin/dspace dsrun org.dspace.content.crosswalk.XSLTDisseminationCrosswalk DataCite [ITEM-HANDLE] [PATH-
TO-OUTPUT-FILE.XML]
```

Identifier Service

The Identifier Service manages the generation, reservation and registration of identifiers within DSpace. You can configure it using the config file located in [dspace]/config/spring/api/identifier-service.xml. In the file you should already find the code to configure DSpace to register DOIs. Just read the comments and remove the comment signs around the two appropriate beans.

After removing the comment signs the file should look something like this (I removed the comments to make the listing shorter):

[dspace]/config/spring/api/identifier-service.xml

```
<!--
    Copyright (c) 2002-2010, DuraSpace.  All rights reserved
    Licensed under the DuraSpace License.

    A copy of the DuraSpace License has been included in this
    distribution and is available at: http://www.dspace.org/license
-->

<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
           http://www.springframework.org/schema/beans/spring-beans-2.5.xsd">

    <bean id="org.dspace.identifier.IdentifierService"
          class="org.dspace.identifier.IdentifierServiceImpl"
          autowire="byType"
          scope="singleton"/>

    <bean id="org.dspace.identifier.DOIIdentifierProvider"
        class="org.dspace.identifier.DOIIdentifierProvider"
        scope="singleton">
        <property name="configurationService"
            ref="org.dspace.services.ConfigurationService" />
        <property name="DOIConnector"
            ref="org.dspace.identifier.doi.DOIConnector" />
    </bean>

    <bean id="org.dspace.identifier.doi.DOIConnector"
        class="org.dspace.identifier.doi.DataCiteConnector"
        scope="singleton">
        <property name='DATACITE_SCHEME' value='https'/>
        <property name='DATACITE_HOST' value='mds.test.datacite.org'/>
        <property name='DATACITE_DOI_PATH' value='/doi/' />
        <property name='DATACITE_METADATA_PATH' value='/metadata/' />
        <property name='disseminationCrosswalkName' value="DataCite" />
    </bean>
</beans>
```

If you use other IdentifierProviders beside the DOIIdentifierProvider there will be more beans in this file.

Please pay attention to configure the property DATACITE_HOST. Per default it is set to the DataCite test server. To reserve real DOIs you will have to change it to mds.datacite.org. Ask your registration agency if you're not sure about the correct address.

DSpace should send updates to DataCite whenever the metadata of an item changes. To do so you have to change the dspace.cfg again. You should remove the comments in front of the two following properties or add them to the dspace.cfg:

[dspace]/config/dspace.cfg

```
event.consumer.doi.class = org.dspace.identifier.doi.DOIConsumer
event.consumer.doi.filters = Item+Modify_Metadata
```

Then you should add 'doi' to the property `event.dispatcher.default.consumers`. After adding it, this property may look like this:

[dspace]/config/dspace.cfg

```
event.dispatcher.default.consumers = versioning, discovery, eperson, doi
```

DOIs using DataCite and Item Level Versioning

If you enabled Item Level Versioning you should enable the `VersionedDOIIdentifierProvider` instead of the `DOIIdentifierProvider`. The `VersionedDOIIdentifierProvider` ensures that newer versions of the same Item gets a DOI looking as the DOI of the first version of and item, extended by a dot and the version number. With DSpace 6 this also became the default for handles if Item Level Versioning is enabled. In the configuration file `[dspace]/config/spring/api/identifier-service.xml` you'll find the possibility to enable the `VersionedDOIIdentifierProvider`. If you want to use versioned DOIS, please comment out the `DOIIdentifierProvider` as only one of both DOIProviders should be enabled at the same time.

Command Line Interface

To make DSpace resistant to outages of DataCite we decided to separate the DOI support into two parts. When a DOI should be generated, reserved or minted, DSpace does this in its own database. To perform registration and/or reservation against the DOI registration agency a job has to be started using the command line. Obviously this should be done by a cron job periodically. In this section we describe the command line interface, in case you ever want to use it manually. In the next section you'll see the cron job that transfers all DOIs designated for reservation and/or registration.

The command line interface in general is documented here: Command Line Operations.

The command used for DOIs is '`doi-organiser`'. You can use the following options:

Option (short)	Option (long)	Parameter	Description
-d	--delete-all		Transmit information to the DOI registration agency about all DOIs that were deleted.
	--delete-doi	DOI	Transmit information to the DOI registration agency that the specified DOI was deleted. The DOI must already be marked for deletion; you cannot use this command to delete a DOI for an existing item.
-h	--help		Print online help.
-l	--list		List all DOIs whose changes were not committed to the registration agency yet.
-q	--quiet		The doi-organiser sends error reports to the mail address configured in the property alert.recipient in dspace.cfg. If you use this option no output should be given to stdout. If you do not use this option the doi-organiser writes information about successful and unsuccessful operations to stdout and stderr. You can find information in dspace.log of course.
-r	--register-all		Transmit information about all DOIs that should be registered.
	--register-doi	DOI | ItemID | handle	If a DOI is marked for registration, you can trigger the registration at the DOI registration agency by this command. Specify either the DOI, the ID of the item, or its handle.
-s	--reserve-all		Transmit to the DOI registration agency information about all DOIs that should be reserved.
	--reserve-doi	DOI | ItemID | handle	If a DOI is marked for registration, you can trigger the registration at the DOI registration agency by this command. Specify either the DOI, the ID of the item, or its handle.
-u	--update-all		If a DOI is reserved for an item, the metadata of the item will be sent to DataCite. This command transmits new metadata for items whose metadata were changed since the DOI was reserved.
	--update-doi	DOI | ItemID | handle	If a DOI needs an update of the metadata of the item it belongs to, you can trigger this update with this command. Specify either the DOI, the ID of the item, or its handle.

Currently you cannot generate new DOIs with this tool. You can only send information about changes in your local DSpace database to the registration agency.

'cron' job for asynchronous reservation/registration

When a DOI should be reserved, registered, deleted or its metadata updated, DSpace just writes this information into its local database. A command line interface is supplied to send the necessary information to the registration agency. This behaviour makes it easier to react to outages or errors while using the API. This information should be sent regularly, so it is a good idea to set up a cron job instead of doing it manually.

There are four commands that should be run regularly:

- Update the metadata of all items that have changed since their DOI was reserved.
- Reserve all DOIs marked for reservation
- Register all DOIs marked for registration
- Delete all DOIs marked for deletion

In DSpace, a DOI can have the state "registered", "reserved", "to be reserved", "to be registered", "needs update", "to be deleted", or "deleted". After updating an item's metadata the state of its assigned DOI is set back to the last state it had before. So, e.g., if a DOI has the state "to be registered" and the metadata of its item changes, it will be set to the state "needs update". After the update is performed its state is set to "to be registered" again. Because of this behaviour **the order of the commands above matters**: the update command must be executed before all of the other commands above.

The cron job should perform the following commands with the rights of the user your DSpace installation runs as:

```
[dspace]/bin/dspace doi-organiser -u -q
[dspace]/bin/dspace doi-organiser -s -q
[dspace]/bin/dspace doi-organiser -r -q
[dspace]/bin/dspace doi-organiser -d -q
```

The doi-organiser sends error messages as email and logs some additional information. The option -q tells DSpace to be quiet. If you don't use this option the doi-organiser will print messages to stdout about every DOI it successfully reserved, registered, updated or deleted. Using a cron job these messages would be sent as email.

In case of an error, consult the log messages. If there is an outage of the API of your registration agency, DSpace will not change the state of the DOIs so that it will do everything necessary when the cron job starts the next time and the API is reachable again.

The frequency the cron job runs depends on your needs and your hardware. The more often you run the cron job the faster your new DOIs will be available online. If you have a lot of submissions and want the DOIs to be available really quickly, you probably should run the cron job every fifteen minutes. If there are just one or two submissions per day, it should be enough to run the cron job twice a day.

To set up the cron job, you just need to run the following command as the *dspace* UNIX user:

```
crontab -e
```

The following line tells cron to run the necessary commands twice a day, at 1am and 1pm. Please notice that the line starting with the numbers is one line, even it it should be shown as multiple lines in your browser.

```
# Send information about new and changed DOIs to the DOI registration agency:
0 1,13 * * *  [dspace]/bin/dspace doi-organiser -u -q ; [dspace]/bin/dspace doi-organiser -s -q ; [dspace]/bin
/dspace doi-organiser -r -q ; [dspace]/bin/dspace doi-organiser -d -q
```

Limitations of DataCite DOI support

Every DSpace installation expects to be the only application that generates DOIs which start with the prefix and the namespace separator you configured. DSpace does not check whether a DOI it generates is reserved or registered already.

That means if you want to use other applications or even more than one DSpace installation to register DOIs with the same prefix, you'll have to use a unique namespace separator for each of them. Also you should not generate DOIs manually with the same prefix and namespace separator you configured within DSpace. For example, if your prefix is 10.5072 you can configure one DSpace installation to generate DOIs starting with 10.5072/papers-, a second installation to generate DOIs starting with 10.5072/data- and another application to generate DOIs starting with 10.5072/results-.

DOIs will be used in addition to Handles. This implementation does not replace Handles with DOIs in DSpace. That means that DSpace will still generate Handles for every item, every collection and every community, and will use those Handles as part of the URL of items, collections and communities.

DSpace currently generates DOIs for items only. There is no support to generate DOIs for Communities and collections yet.

When using DSpace's support for the DataCite API probably not all information would be restored when using the AIP Backup and Restore (see DS-1836). The DOIs included in metadata of Items will be restored, but DSpace won't update the metadata of those items at DataCite anymore. You can even get problems when minting new DOIs after you restored older once using AIP.

Configure DSpace to use EZID service for registration of DOIs

The EZID IdentifierProvider operates synchronously, so there is much less to configure. You will need to un-comment the `org.dspace.identifier.EZIDIdentifierProvider` bean in `config/spring/api/identifier-service.xml` to enable DOI registration through EZID.

In `config/dspace.cfg` you will find a small block of settings whose names begin with `identifier.doi.ezid`. You should uncomment these properties and give them appropriate values. Sample values for a test account are supplied.

name	meaning
identifier.doi.ezid.shoulder	The "shoulder" is the DOI prefix issued to you by the EZID service. DOIs minted by this instance of DSpace will be the concatenation of the "shoulder" and a locally unique token.
identifier.doi.ezid.user identifier.doi.ezid.password	The username and password by which you authenticate to EZID.
identifier.doi.ezid.publisher	You may specify a default value for the required `datacite.publisher` metadatum, for use when the Item has no publisher.
crosswalk.dissemination. DataCite.publisher	Should match identifier.doi.ezid.publisher.
crosswalk.dissemination. DataCite.hostingInstitution	Name of the hosting institution. If not configured, it will be set to the value of crosswalk.dissemination.DataCite. publisher.
crosswalk.dissemination. DataCite.dataManager	Name of the data manager. If not configured, it will be set to the value of crosswalk.dissemination.DataCite. publisher.

Back in `config/spring/api/identifier-service.xml` you will see some other configuration of the `EZIDIdentiferProvider` bean. In most situations, the default settings should work well. But, here's an explanation of options available:

- *EZID Provider / Registrar settings:* By default, the EZIDIdentifierProvider is configured to use the CDLib provider (ezid.cdlib.org) in the `EZID_SCHEME`, `EZID_HOST` and `EZID_PATH` settings. In most situations, the default values should work for you. However, you may need to modify these values (especially the `EZID_HOST`) if you are registered with a different EZID provider. In that situation, please check with your provider for valid "host" and "path" settings. If your provider provides EZID service at a particular path on its host, you may set that in `EZID_PATH`.
    - NOTE: As of the writing of this documentation, the default CDLib provider settings should also work for institutions that use Purdue (ezid.lib.purdue.edu) as a provider. Currently, Purdue and CDLib currently share the same infrastructure, and both `ezid.cdlib.org` and `ezid.lib.purdue.edu` point to the same location.
- *Metadata mappings:* You can alter the mapping between DSpace and EZID metadata, should you choose. The `crosswalk` property is a map from DSpace metadata fields to EZID fields, and can be extended or changed. The `key` of each `entry` is the name of an EZID metadata field; the `value` is the name of the corresponding DSpace field, from which the EZID metadata will be populated.
- *Crosswalking / Transforms:* You can also supply transformations to be applied to field values using the `crosswalkTransform` property. Each `key` is the name of an EZID metadata field, and its `value` is the name of a Java class which will convert the value of the corresponding DSpace field to its EZID form. The only transformation currently provided is one which converts a date to the year of that date, named `org.dspace.identifier.ezid.DateToYear`. In the configuration as delivered, it is used to convert the date of issue to the year of publication. You may create new Java classes with which to supply other transformations, and map them to metadata fields here. If an EZID metadatum is not named in this map, the default mapping is applied: the string value of the DSpace field is copied verbatim.

Limitations of EZID DOI support

DOIs will be used in addition to Handles. This implementation does not replace Handles with DOIs in DSpace. That means that DSpace will continue to generate Handles for every item, every collection and every community, and will use those Handles as part of the URL of items, collections and communities.

Currently, the EZIDIdentifierProvider has a known issue where it stores its DOIs in the `dc.identifier` field, instead of using the `dc.identifier.uri` field (which is the one used by DataCite DOIs and Handles). See DS-2199 for more details. This will be corrected in a future version of DSpace.

DSpace currently generates DOIs for items only. There is no support to generate DOIs for Communities and Collections yet.

JSPUI specific configurations

You can configure whether the JSPUI should show DOIs or handles on item frontdoors. Heading an item frontdoor there is an informational note containing a Persistent Identifier and the request to use it when one wants to refer to this item. By setting the property webui.preferred.identifier to doi in dspace.cfg, you can configure the JSPUI to use DOIs instead of handles which are used by default. This property also controls which Persistent Identifiers are used in the Version History that is shown if Item Level Versioning is used and version history is enabled.

Further more you can configure whether DOIs should contain a doi: prefix or not in the version history. The property webui.identifier.strip-prefixes in dspace.cfg controls this. By default the doi: prefix is stripped (not shown).

Adding support for other Registration Agencies

If you want DSpace to support other registration agencies, you just have to write a Java class that implements the interface DOIConnector ([dspace-source]/dspace-api/src/main/java/org/dspace/identifier/doi/DOIConnector.java). You might use the DataCiteConnector ([dspace-source]/dspace-api/src/main/java/org/dspace/identifier/doi/DataCiteConnector.java) as an example. After developing your own DOIConnector, you configure DSpace as if you were using the DataCite API directly. Just use your DOIConnector when configuring the IdentifierService instead of the DataCiteConnector.