

Migrating EPrints data into DSpace

1 About this document

Author

Corey Wallis, RUBRIC Technical Officer

Purpose

This technical report outlines how to use the EPrints data migration scripts to ingest MARC records into a DSpace repository.

Audience

RUBRIC Project Partners and other users of EPrints and DSpace

Requirements

A running instance of DSpace

An XML export from an EPrints repository

Python 2.4 installed on the system to run the migration

The lxml2 library, including the Python bindings, installed on the system to run the migration

The libxslt library, including the Python bindings, installed on the system to run the migration

References

Official DSpace website

<http://www.dspace.org/>

Official DSpace Item Importer documentation

<http://dspace.org/technology/system-docs/application.html#itemimporter>

Official DSpace Dublin Core specification

<http://www.dspace.org/technology/metadata.html>

Official EPrints website

<http://www.eprints.org/software/>

Official EPrints documentation on the export_xml command

http://www.eprints.org/documentation/tech/php/export_xml.php

Official Python website

<http://www.python.org/>

Official lxml2 website

<http://xmlsoft.org/python.html>

Official libxslt website

<http://xmlsoft.org/XSLT/python.html>

Official py.test tool and library website

<http://codespeak.net/py/current/doc/test.html>

Official utf-x website

<http://utf-x.sourceforge.net/>

Official Subversion website

<http://subversion.tigris.org/>

Notes

The EPrints to DSpace migration script has been developed on a Linux based system. Python is a cross platform programming language and therefore the scripts should also run under Microsoft Windows, and the OSX operating systems. This has not been tested.

The installation of the Python programming language, the libxml2 and libxslt libraries, including Python bindings, is outside the scope of this technical report. Many Linux distributions, such as Ubuntu, will have these already installed.

The EPrints to DSpace metadata transformation was developed using sample data from the University of Southern Queensland. Therefore adjustments may need to be made to the XSL transformation to take into account other institutions requirements.

Due to the modular design of the EPrints to DSpace data migration, these adjustments can be made without the need to modify the Python script, or the other supporting files. If modifications are to be made it is strongly suggested that a copy of the development files be checked out via subversion. This will provide you with the utf-x tests used to develop the original files and provide a basis for customisation using the same utf-x framework.

2 Background Information

A component of the work undertaken at RUBRIC-Central is the development of various data migration strategies. These strategies are designed to assist RUBRIC Project Partners to migrate data into, and out of, various systems. The data migrations specifically target the three institutional repository solutions under consideration as part of the project.

Interest was expressed in being able to migrate records, from an Eprints repository, into a DSpace repository. This technical report, and the associated Python scripts, comprise the strategy to achieve this goal.

The Python script creates a DSpace simple archive that can then be ingested into a DSpace repository. The output of the script has been tested using DSpace version 1.4 alpha. It is possible that the output can be ingested into other versions of DSpace, at the time of writing this had not been tested.

The Python scripts have been developed using a unit testing approach using the testing framework provided by the py.test tool and library. More information about the library is available at <http://codespeak.net/py/current/doc/test.html>. These scripts have been developed using Python version 2.4, and may work with earlier versions. However this has not been tested.

The conversion of the EPrints metadata into the Dublin Core supported by DSpace is achieved using XSL transformations. The stylesheets have been developed using a unit testing approach using the framework provided by the utf-x suite. More information about the utf-x suite is available at <http://utf-x.sourceforge.net/>.

3 The Python Scripts

The data migration work is carried out by a script written in the Python programming language. The following files make up the script used in the data migration.

config.xml

The configuration file defines options used by the **eprints_to_dspace.py** script

eprints_to_dspace.py

The Python script that does all of the work

./xsl/eprints_to_dspace.xsl

The XSL transformation used to convert the EPrints metadata into DSpace compliant Dublin Core

./xsl/ASRC.xml

An XML file containing all of the ASRC subject codes. It is used by the **eprints_to_dspace.xsl** file to replace the ASRC subject number with the complete description. For more information on this file see the RUBRIC Technical Report: ASRC Subject codes for DSpace

4 Downloading the Python Script

All of the data migration scripts, and associated code libraries, modules and files, are made available via a publicly accessible website at the following URL.

<https://rubric-central.usq.edu.au/svn/Public/code>

It is possible to download files from this website in two different ways. The first is via a standard Internet browser. The second is via a subversion client.

4.1 Download the Distribution Package

The Python script, and associated files have been packaged for ease of download as a tar.gz file. The file is available via the following URL:

https://rubric-central.usq.edu.au/svn/Public/code?legacy_code/eprints-to-dspace/eprints_to_dspace.tar.gz

The procedure for configuring, and using, the scripts is outlined in section 5 of this technical report.

4.2 Download the Python Script via Subversion

If you have the subversion client installed you can download the Python scripts, test files, and other files used during development. The URL that you will need to check out is as follows:

https://rubric-central.usq.edu.au/svn/Public/code/legacy_code/eprints-to-dspace/development/

5 How to Migrate the EPrints data

The following sections of this technical report outline the procedure for using the Python script to migrate the EPrints data into a DSpace simple archive and ingest this archive into a DSpace repository.

It is assumed that you have installed Python, the libxml2 and libxslt libraries, including Python bindings. Specific installation instructions for these components is outside the scope of this technical report.

5.1 Installing the Python Script

1. Download the **eprints_to_dspace.tar.gz** file from the URL outlined in section 4.11. https://rubric-central.usq.edu.au/svn/Public/code/legacy_code/eprints-to-dspace/eprints_to_dspace.tar.gz

2. Extract the contents of the file

```
tar -xzf ~/eprints-to-dspace.tar.gz
```

3. The Python script, and supporting files, will be extracted into the following directory

```
~/eprints-to-dspace
```

5.2 Configuring the Python Script

The configuration options for the Python script is stored in the **config.xml** file. Each option is outlined below, adjust the values to your local configuration.

doXSLTransformation

If set to True, this option instructs the script to conduct the XSL transformation and convert the EPrints metadata into DSpace compliant Dublin Core

doCreateArchive

If set to True, this option instructs the script to create the DSpace simple archive, using the output of the XSL transformation

xslLocation

The location of the XSL transformation used to convert the EPrints metadata into DSpace compliant Dublin Core. Leave this configuration at the default value

xslOutputLocation

This configuration option defines where the output of the XSL transformation should be stored. If the **createDirectory** attribute is set to **True**, the directory will be created if it does not exist, and deleted then recreated if it does exist.

eprintsMetadataLocation

This configuration option defines where the XML file created by the EPrints **xml_export** command is stored

dspaceLocation

This configuration option defines where the DSpace simple archive will be created. If the **createDirectory** attribute is set to **True**, the directory will be created if it does not exist, and deleted then recreated if it does exist.

Once all of the configuration changes have been made, save the file and exit out of your editor.

5.3 Creating the DSpace Simple Archive

The creation of the DSpace Simple Archive is a two step process. First it is necessary to convert the Eprints metadata into DSpace compliant Dublin Core. The second step takes the Dublin Core and creates a DSpace Simple Archive.

5.3.1 Converting the EPrints metadata

1. Ensure the **doXSLTransformation** configuration option is set to **True**
2. Ensure the **doCreateArchive** configuration option is set to **False**
3. Ensure all of the other configuration options are correct
4. Invoke the script using the following command

```
./eprints_to_dspace.py
```

5. Upon completion a series of XML files will be created in the directory specified by the **xslOutputLocation** configuration option

5.3.2 Creating the DSpace Simple Archive

1. Ensure the **doXSLTransformation** configuration option is set to **False**
2. Ensure the **doCreateArchive** configuration option is set to **True**
3. Ensure all of the other configuration options are correct
4. Invoke the script using the following command

```
./eprints_to_dspace.py
```

5. Upon completion a the DSpace simple archive will be created in the directory specified by the **dspaceLocation** configuration option

5.3.3 Creating the DSpace Simple Archive in one step

1. Ensure the **doXSLTransformation** configuration option is set to **True**
2. Ensure the **doCreateArchive** configuration option is set to **True**
3. Ensure all of the other configuration options are correct
4. Invoke the script using the following command

```
./eprints_to_dspace.py
```

5. Upon completion a series of XML files will be created in the directory specified by the **xslOutputLocation** configuration option
6. Upon completion a the DSpace simple archive will be created in the directory specified by the **dspaceLocation** configuration option

5.4 Ingesting the DSpace Simple Archive

Ingesting the DSpace simple archive is a three step process. The first is to run a test ingest, if successful continue with a true ingest, and finally re-index the repository to ensure the indexes are updated with the new objects.

If for any reason it is necessary to delete the imported objects, section 5.4.4 outlines this process.

Before ingesting the items it is necessary to identify the handle of the collection that these items are to be associated with. To achieve this, complete the following procedure:

1. Visit the DSpace homepage in your favourite Internet browser
2. Click on the **Communities & Collections** link in the browse menu
3. Hover the mouse over the link for the collection you wish to import into
4. Determine the handle for the collection by examining the URL displayed in the status bar of your Internet browser. The numbers after the word **handle** are the handle for this collection.

5.4.1 Ingesting the DSpace Simple Archive – Test

1. If necessary copy the DSpace Simple Archive to the server running DSpace
2. Ensure the DSpace Simple Archive is accessible to the **dspace** user
3. Change to the **dspace** user
4. Navigate to the **bin** directory of the DSpace installation. For example on RUBRIC systems this directory is at the following location

```
/usr/local/dspace/bin
```

5. Invoke the following command replacing:
 - [eperson] with the email address associated with a DSpace eperson with sufficient privileges to add items to the repository
 - [collection] with the collection handle identified in section 5.5
 - [source] with the location of the DSpace simple archive
 - [map] with a suitable location for the map file

```
./dsrun org.dspace.app.itemimport.ItemImport -add  
--eperson=[eperson] -collection=[collection]  
--source=[source] --mapfile=[map] --test
```
6. If the ingest is successful the DSpace utility will display an appropriate success message.
7. If the ingest fails for any reason, examine the error message and take the appropriate action to resolve the error condition.

5.4.2 Ingesting the DSpace Simple Archive – Live

1. Ensure the DSpace Simple Archive is accessible to the **dspace** user
2. Change to the **dspace** user
3. Navigate to the **bin** directory of the DSpace installation. For example on RUBRIC systems this directory is at the following location

```
/usr/local/dspace/bin
```
4. Invoke the following command replacing:
 - [eperson] with the email address associated with a DSpace eperson with sufficient privileges to add items to the repository
 - [collection] with the collection handle identified in section 5.5
 - [source] with the location of the DSpace simple archive
 - [map] with a suitable location for the map file

```
./dsrun org.dspace.app.itemimport.ItemImport --add  
--eperson=[eperson] --collection=[collection]  
--source=[source] --mapfile=[map]
```
5. If the ingest is successful the DSpace utility will display an appropriate success message.
6. If the ingest fails for any reason, examine the error message and take the appropriate action to resolve the error condition.

5.4.3 Re-Indexing the DSpace repository

1. To minimise impact to users of the system, ensure the DSpace repository isn't under any heavy load
2. Ensure you are in the **bin** directory of the DSpace installation. For example on RUBRIC systems this directory is at the following location

```
/usr/local/dspace/bin
```

3. Invoke the following command

```
./index-all
```

1. Visit the DSpace homepage in your favourite Internet browser
2. Click on the **Communities & Collections** link in the browse menu
3. Click on the link for the collection that the new items were ingested into
4. Explore the collection to ensure the ingest completed successfully

5.4.4 Deleting the Imported Items

If it is necessary to remove the items from the DSpace repository, complete the following procedure.

1. Ensure the map file created in section 5.5.2 is still available
2. Ensure the map file is accessible to the **dspace** user
3. Change to the **dspace** user
4. Navigate to the **bin** directory of the DSpace installation. For example on RUBRIC systems this directory is at the following location

```
/usr/local/dspace/bin
```

5. Invoke the following command replacing:

- [eperson] with the email address associated with a DSpace eperson with sufficient privileges to add items to the repository
- [map] with the location of the map file created in section 5.5.2

```
./dsrun org.dspace.app.itemimport.ItemImport --delete
```

```
--eperson=[eperson] --mapfile=[map]
```

6. If the deletion is successful the DSpace utility will display an appropriate success message
7. If the delete fails for any reason, examine the error message and take the appropriate action to resolve the error condition