



Data migration

CERN-UNESCO School on Digital Libraries, Nov. 29, 2016  
Kenneth Hole



# Agenda

Give you the necessary tools to manipulate and import data to your digital library system.

## How?

- Introduction to MARCXML
- Introduction to Reese's MarcEdit
- Introduction to lxml python library
- Demonstrate a simple script

# INTRODUCTION TO MARCXML

```
<record>
  <datafield tag="035" ind1=" " ind2=" ">
    <subfield code="a">ocn650875228</subfield>
  </datafield>
  <datafield tag="040" ind1=" " ind2=" ">
    <subfield code="a">GPO</subfield>
    <subfield code="c">GPO</subfield>
    <subfield code="d">MvI</subfield>
  </datafield>
  </datafield>
  <datafield tag="100" ind1="1" ind2=" ">
    <subfield code="a">Snyder, Christopher A.</subfield>
  </datafield>
  <datafield tag="245" ind1="1" ind2="0">
    <subfield code="a">Gas turbine characteristics for a large civil tilt-rotor (LCTR)</subfield>
    <subfield code="h">[electronic resource] /</subfield>
    <subfield code="c">Christopher A. Snyder, Douglas R. Thurman ; prepared for the 65th Annual Forum and
  </datafield>
  <datafield tag="260" ind1=" " ind2=" ">
    <subfield code="a">Cleveland, Ohio :</subfield>
    <subfield code="b">National Aeronautics and Space Administration, Glenn Research Center,</subfield>
    <subfield code="c">[2010]</subfield>
  </datafield>
</record>
```

## Case Study Walkthrough

**Use case:**

**Convert Springer OA e-book records to fit Invenio standard.**

# Steps

Step 1: Download file

Step 2: Convert from .mrc → MARCXML with MarcEdit

Step 3: Convert data using the lxml Python library

Step 4: Upload records

# Download File

Search



[Home](#) [Subjects](#) [Services](#) [Products](#) [Springer Shop](#) [About us](#)

## Services for Librarians

» [Springer @ Your Library](#)

» [MARC records](#)

[MARC record videos](#)

[Springer MARC records & eBook title lists](#)

[Free OCLC MARC records](#)

[Springer Reference records](#)

[Contact MARC helpdesk](#)

[Catalog update newsletter](#)

## Springer MARC records - title list download tool

Download your Springer eBooks title list, the title list of SpringerProtocols as well as the MARC records tailored to your collection(s).

The URLs linking to all Palgrave titles in the MARC records available here point to Palgraveconnect.com. Springer will re-register those shortly to SpringerLink.

---

► [Springer's metadata policy:](#)

---

<https://www.springer.com/gp/librarians/marc/marc-records-tool>

## Convert from .mrc → MARCXML with MarcEdit

0 1 9 3 6 n a m a 2 2 0 0 5 0 5 5 i  
4500001001800000003000900018005001700027007001500044008004100059020003700100024003100137050001  
7001680720018001850720023002030820012002261000031002382450075002692640077003443000033004213360  
0260045433700260048033800360050634700240054250600160056652002800058265000160086265000330087865  
0003000911650002400941650003600965650003301001650001601034650003501050650003901085650003501124  
6500028011596500019011876500024012067100034012307730020012647760036012848560044013209120014013  
64950005201378978-0-230-10977-3DE-He21320160316141845.0cr nn 008mamaa151013s2010 xxu| s ||| |O|  
eng d a97802301097739978-0-230-10977-37 a10.1057/97802301097732doi 4aPN441-1009.5 7aDSBH52bicssc  
7aLIT0240002bisacsh04a8092231 aBrown, J. Andrew.eauthor.10aCyborgs in Latin America[electronic  
resource] /cby J. Andrew Brown. 1aNew York :bPalgrave Macmillan US :blmprint: Palgrave Macmillan,c2010.  
aXI, 212 p.online resource. atextbtxt2rdacontent acomputerbc2rdamedia aonline resourcebcr2rdacarrier  
atext filebPDF2rda0 aOpen Access aA PDF version of this book is available for free in open access via the  
OAPEN Library platform, www.oapen.org . Cyborgs in Latin America explores the ways cultural expression in  
Latin America has grappled with the changing relationships between technology and human identity.  
0aLiterature. 0aCulturexStudy and teaching. 0aEthnologyxLatin America. 0aPhilosophy of mind. 0aPhilosophy  
and social sciences. 0aSocial sciencesxPhilosophy.14aLiterature.24aPostcolonial/World Literature.  
24aPhilosophy of the Social Sciences.24aRegional and Cultural Studies.24aLatin American Culture.24aSocial  
Theory.24aPhilosophy of Mind.2 aSpringerLink (Online service)0 tSpringer eBooks08iPrinted  
edition:z978134928835940uhttp://dx.doi.org/10.1057/9780230109773 aZDB-2-PCO aPalgrave Literature  
C o l l e c t i o n ( S p r i n g e r - 4 1 1 4 1 ) 0 2 5 1 5 n a m a 2 2 0 0 5 1 7 5 i  
4500001001800000003000900018005001700027007001500044008004100059020003700100024003100137050001  
1001680720016001790720023001950820012002181000032002302450131002622640077003933000033004703360  
0260050333700260052933800360055534700240059150600160061552007910063165000230142265000220144565

# Convert from .mrc → XML with MarcEdit

```
<marc:collection>
<marc:record>
<marc:leader>01936nam a22005055i 4500</marc:leader>
<marc:controlfield tag="001">978-0-230-10977-3</marc:controlfield>
<marc:controlfield tag="003">DE-He213</marc:controlfield>
<marc:controlfield tag="005">20160316141845.0</marc:controlfield>
<marc:controlfield tag="007">cr nn 008mamaa</marc:controlfield>
<marc:controlfield tag="008">151013s2010 xxu| s ||| 0|eng d</marc:controlfield>
<marc:datafield tag="020" ind1=" " ind2=" ">
  <marc:subfield code="a">9780230109773</marc:subfield>
  <marc:subfield code="9">978-0-230-10977-3</marc:subfield>
</marc:datafield>
<marc:datafield tag="024" ind1="7" ind2=" ">
  <marc:subfield code="a">10.1057/9780230109773</marc:subfield>
  <marc:subfield code="2">doi</marc:subfield></marc:datafield>
<marc:datafield tag="050" ind1=" " ind2="4">
  <marc:subfield code="a">PN441-1009.5</marc:subfield>
</marc:datafield>
<marc:datafield tag="072" ind1=" " ind2="7">
  <marc:subfield code="a">DSBH5</marc:subfield>
  <marc:subfield code="2">bicssc</marc:subfield>
</marc:datafield>
<marc:datafield tag="072" ind1=" " ind2="7">
  <marc:subfield code="a">LIT024000</marc:subfield>
  <marc:subfield code="2">bisacsh</marc:subfield>
</marc:datafield>
<marc:datafield tag="082" ind1="0" ind2="4">
  <marc:subfield code="a">809</marc:subfield>
  <marc:subfield code="2">23</marc:subfield>
</marc:datafield>
<marc:datafield tag="100" ind1="1" ind2=" ">
  <marc:subfield code="a">Brown, J. Andrew.</marc:subfield>
  <marc:subfield code="e">author.</marc:subfield>
</marc:datafield>
<marc:datafield tag="245" ind1="1" ind2="0">
  <marc:subfield code="a">Cyborgs in Latin America</marc:subfield>
  <marc:subfield code="h">[electronic resource] </marc:subfield>
  <marc:subfield code="c">by J. Andrew Brown.</marc:subfield>
</marc:datafield>
<marc:datafield tag="264" ind1=" " ind2="1">
  <marc:subfield code="a">New York :</marc:subfield>
  <marc:subfield code="b">Palgrave Macmillan US :</marc:subfield>
  <marc:subfield code="b">Imprint: Palgrave Macmillan,</marc:subfield>
  <marc:subfield code="c">2010.</marc:subfield>
</marc:datafield>
```



# Convert from .mrc → MARCXML with MarcEdit

The image shows two overlapping windows from the MarcEdit application. The background window, titled "MarcEdit By Terry Reese", displays a grid of tool icons: "MARC Tools" (wrench and hammer), "MARCJoin" (green square with white arrow), "MarcEditor" (orange square with pencil), "MARCSPlit" (red square with white arrows), "RDA Helper" (red square with white text), "MARCValidator" (purple square with pencil and checkmark), and "Z39.50 Client" (black square with white magnifying glass). A gear icon is visible at the bottom right of this window. The foreground window, titled "MARC Tools", is a configuration panel for the "MARC Tools" tool. It features an "Input File" field with the text "Springer\_MARC\_20161102\_124237.mrc" and a folder selection icon. Below it is a "Save File" field with the text "Springer\_MARC\_20161102\_124237.xml" and another folder selection icon. A "Function List" dropdown menu is set to "MARC=>MARCXML". There are two checkboxes: "Translate to MARC-8" and "Translate to UTF-8", both of which are currently unchecked. At the bottom of the panel is a "Results" text area. On the right side of the "MARC Tools" window, there are three buttons: "Execute", "Edit Records", and "Close".

Add URL

# Programming techniques

Transformation Process	Programming Technique
Import required Python library	Import
Read XML file and parse into an objects	Read XML and parse it.
Remove data field 001 and 005	Iterate over the object and remove content
Add collection name in data field 980	Iterate over list and add new data field
Write data to new XML file	Create, write and save XML file

## Step 1: Import python library

# Import the Python library needed to read and process XML

```
import lxml.etree as ET
```

## Step 2: Read XML and parse it

```
# Read XML and parse it into an object  
xml = ET.parse("springer_OA_input.xml")
```

```
# Get to the root level (collection)  
root = xml.getroot()
```

```
<Element collection at 0x1022dc7e8>
```

## Step 3: Iterate over list and remove content

*# Iterate over all records in the collection*

```
for record in root:
```

```
    for field in record:
```

```
        if field.tag == 'controlfield':
```

```
            if field.attrib['tag'] == "001":
```

```
                record.remove(field)
```

```
            if field.attrib['tag'] == "005":
```

```
                record.remove(field)
```

## Step 3: Iterate over list and remove content

```
for record in root:  
    # Iterate over all fields in the record  
    for field in record:  
        if field.tag == 'controlfield':  
            if field.attrib['tag'] == "001":  
                record.remove(field)  
            if field.attrib['tag'] == "005":  
                record.remove(field)
```

## Step 3: Iterate over list and remove content

```
for record in root:  
    for field in record:  
        # If the field tag equals the string 'controlfield',  
        continue  
        if field.tag == 'controlfield':  
            if field.attrib['tag'] == "001":  
                record.remove(field)  
            if field.attrib['tag'] == "005":  
                record.remove(field)
```

## Step 3: Iterate over list and remove content

```
for record in root:
    for field in record:
        if field.tag == 'controlfield':
            # If the controlfield attribute equals '001', continue
            if field.attrib['tag'] == "001":
                record.remove(field)
            if field.attrib['tag'] == "005":
                record.remove(field)
```



## Step 3: Iterate over list and remove content

```
for record in root:
    for field in record:
        if field.tag == 'controlfield':
            if field.attrib['tag'] == "001":
                # Remove the data field from the record
                record.remove(field)
            if field.attrib['tag'] == "005":
                record.remove(field)
```

## Step 3: Iterate over list and remove content

```
for record in root:
    for field in record:
        if field.tag == 'controlfield':
            if field.attrib['tag'] == "001":
                record.remove(field)
            # If the controlfield attribute equals '005', continue
            if field.attrib['tag'] == "005":
                record.remove(field)
```

## Step 3: Iterate over list and remove content

```
for record in root:  
    for field in record:  
        if field.tag == 'controlfield':  
            if field.attrib['tag'] == "001":  
                record.remove(field)  
            if field.attrib['tag'] == "005":  
                # Remove the data field from the record  
                record.remove(field)
```

## Step 4: Iterate over list and add content

*# Iterate over all records in the collection*

**for** record **in** root:

new\_980\_field = ET.Element('datafield',

attrib={'tag': '980', 'ind1': ' ', 'ind2': ' '})

new\_subfield = ET.Element('subfield', attrib={'code': 'a'})

new\_subfield.text = 'BIB'

new\_980\_field.append(new\_subfield)

record.append(new\_980\_field)

## Step 4: Iterate over list and add content

**for** record **in** root:

*# Create a new data field element with attributes and save it as "new\_980\_field".*

```
new_980_field = ET.Element('datafield',  
                           attrib={'tag': '980', 'ind1': ' ', 'ind2': ' '})  
new_subfield = ET.Element('subfield', attrib={'code': 'a'})  
new_subfield.text = 'BIB'  
new_980_field.append(new_subfield)  
record.append(new_980_field)
```

## Step 4: Iterate over list and add content

```
for record in root:
    new_980_field = ET.Element('datafield',
                               attrib={'tag': '980', 'ind1': ' ', 'ind2': ' '})
    #Create a new subfield element with attribute and save it as "new_subfield".
    new_subfield = ET.Element('subfield', attrib={'code': 'a'})
    new_subfield.text = 'BIB'
    new_980_field.append(new_subfield)
    record.append(new_980_field)
```

## Step 4: Iterate over list and add content

```
for record in root:
    new_980_field = ET.Element('datafield',
                               attrib={'tag': '980', 'ind1': ' ', 'ind2': ' '})
    new_subfield = ET.Element('subfield', attrib={'code': 'a'})
    # Add the string 'BIB' as text to the subfield
    new_subfield.text = 'BIB'
    new_980_field.append(new_subfield)
    record.append(new_980_field)
```

## Step 4: Iterate over list and add content

```
for record in root:
    new_980_field = ET.Element('datafield',
                               attrib={'tag': '980', 'ind1': ' ', 'ind2': ' '})
    new_subfield = ET.Element('subfield', attrib={'code': 'a'})
    new_subfield.text = 'BIB'
    # Append subfield to data field.
    new_980_field.append(new_subfield)
    record.append(new_980_field)
```



## Step 4: Iterate over list and add content

```
for record in root:
    new_980_field = ET.Element('datafield',
                               attrib={'tag': '980', 'ind1': ' ', 'ind2': ' '})
    new_subfield = ET.Element('subfield', attrib={'code': 'a'})
    new_subfield.text = 'BIB'
    new_980_field.append(new_subfield)
    # Append data field to record.
    record.append(new_980_field)
```

## Step 5: Write data to new XMLfile

```
# Create and open a new XML file called  
"springer_OA_output"  
xmlfile = open('springer_OA_output.xml', 'w')  
xmlfile.write(ET.tostring(root, pretty_print=True,  
encoding='UTF-8'))  
xmlfile.close()
```

## Step 5: Write data to new XMLfile

```
xmlfile = open('springer_OA_output.xml', 'w')  
# Convert root to a string, encode it to UTF-8 and make it  
look pretty!  
Write the string to the "springer_OA_output.xml" file  
xmlfile.write(ET.tostring(root, pretty_print=True,  
encoding='UTF-8'))  
xmlfile.close()
```

## Step 5: Write data to new XMLfile

```
xmlfile = open('springer_OA_output.xml', 'w')
xmlfile.write(ET.tostring(root, pretty_print=True,
encoding='UTF-8'))
# Close the file
xmlfile.close()
```

**Run script!**

## Upload records

## Other applications

- Search for specific data
- Find duplicates
- Create statistics
- More?

# Documentation

Learning Python

<https://www.codecademy.com/tracks/python>

Lxml library

<http://lxml.de/>



The logo consists of the word "TIND" in a white, sans-serif font, centered within a white rectangular border. The border has a slight inward curve at the bottom. The background is a solid blue color.

TIND

[www.tind.io](http://www.tind.io)