Transformation of BHL Europe data using Pentaho Data Integration

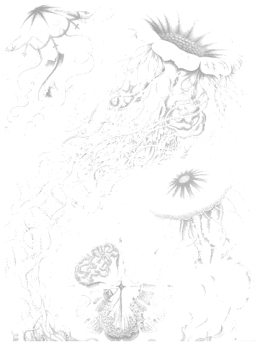
Biodiversity Heritage Library Europe

co-funded by the Community programme eContent*plus*

europeana project group member

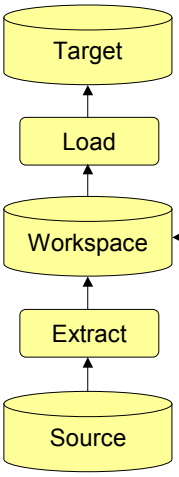
Outline:

- Introduction to Pentaho Data Integration
- Transforming Data from Relational Databases
- Transforming Data from XML Input



Introduction to Pentaho Data Integration (1)

- Part of Pentaho Business Intelligence Suite
- ETL Tool
- Open Source

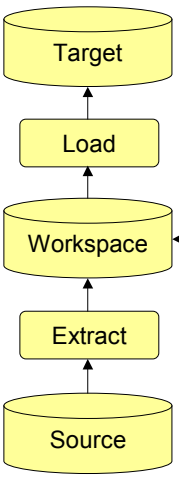


Introduction to Pentaho Data Integration (2)

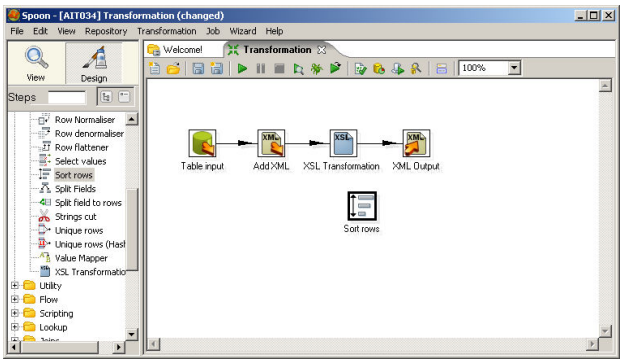
Output steps

Transform steps

Input steps

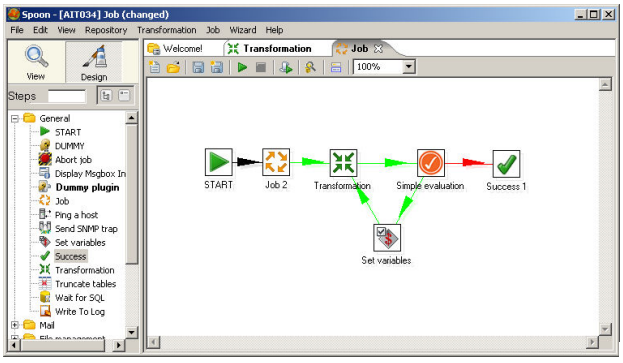


Introduction to Pentaho Data Integration (3)



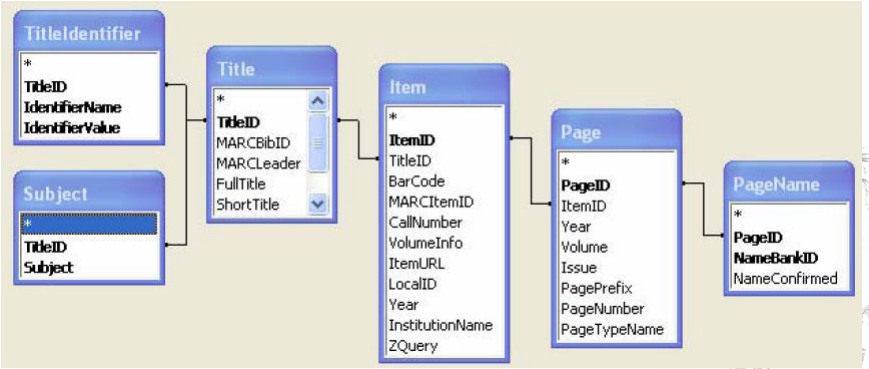
Steps are created and connected via Drag and Drop.

Introduction to Pentaho Data Integration (4)



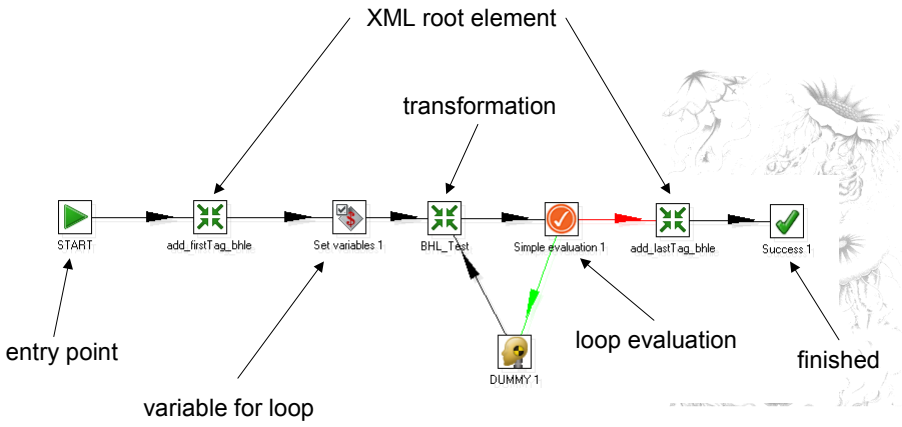
- Jobs control the workflow
- Other jobs and transformations can be called
- Conditional evaluations are possible (loops)

Transforming Data from Relational Databases

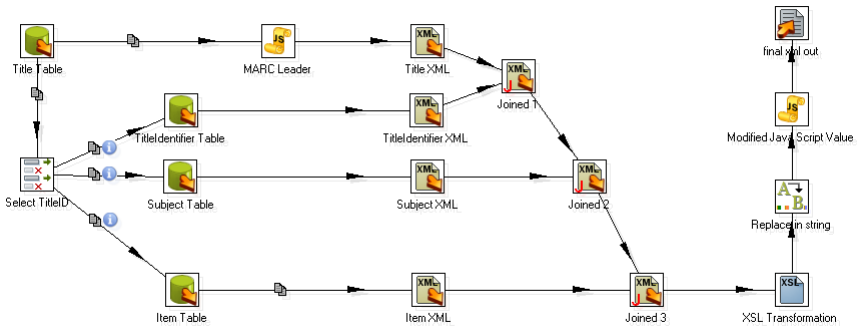


Database schema (Export) of Missouri Botanical Garden (MOBOT)

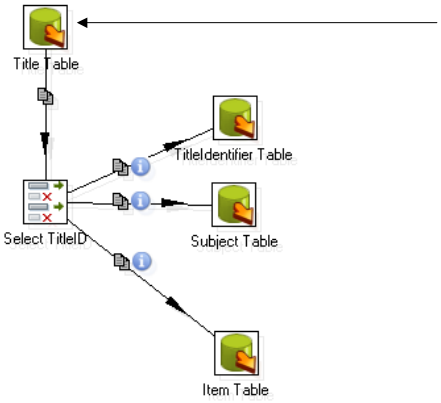
Missouri Botanical Garden (MOBOT) - Job

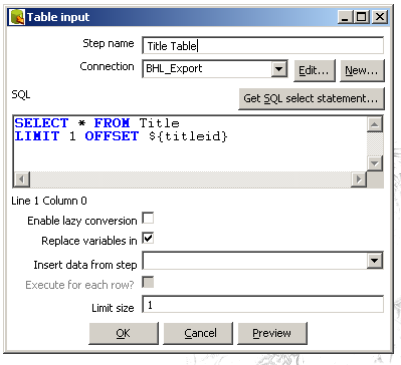


Missouri Botanical Garden (MOBOT) - Transformation



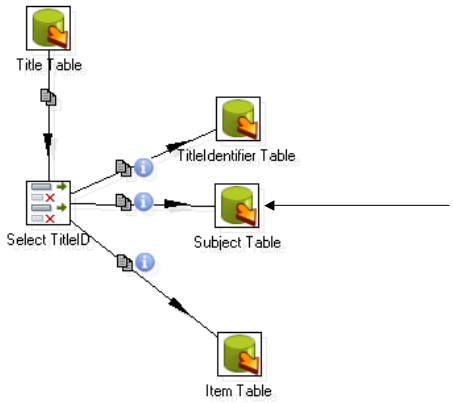
Getting Data from a Relational Database (1)

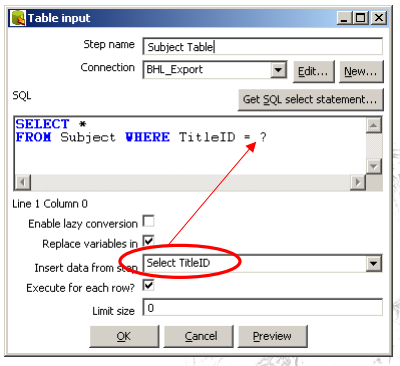




Gets one row depending on loop variable of the job.

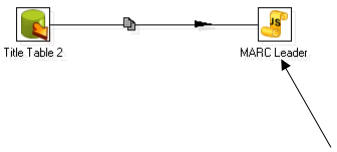
Getting Data from a Relational Database (2)





Gets rows depending on the TitleID of the current title.

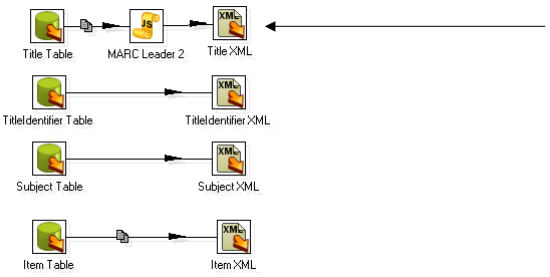
Special Treatment of Fields



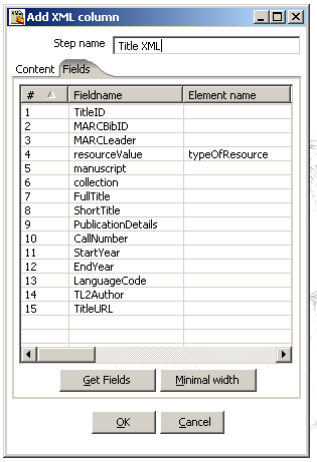
JavaScript can handle special fields, e.g. extracting information from the MARCLeader.

```
Script 1
//Script here

var marcleader= MARCLeader;
var resourceValue;
var manuscript;
var collection;

var pos_six = MARCLeader.charAt(6);
var pos_seven = MARCLeader.charAt(7);

  switch (pos_six) {
    case "a": resourceValue = "text";
              break;
    case "t": resourceValue = "text";
              break;

    case "e": resourceValue = "cartographic";
              break;

    case "f": resourceValue = "cartographic";
              break;

    case "c": resourceValue = "notated music";
              break;

    case "d": resourceValue = "notated music";
              break;
```

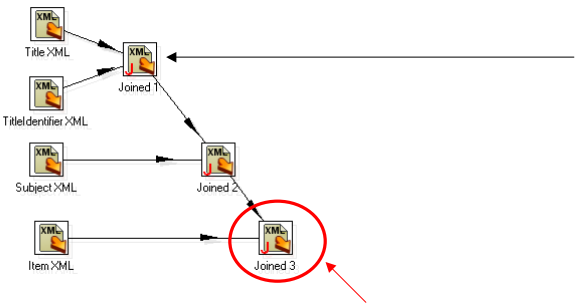
Creating XML Parts



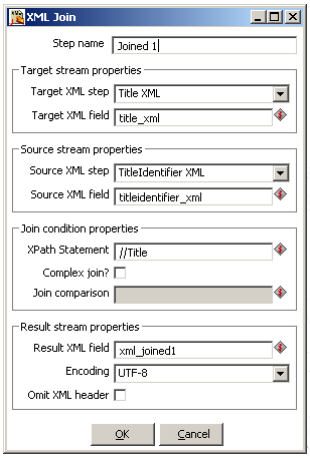
XML data is created with information from each table.



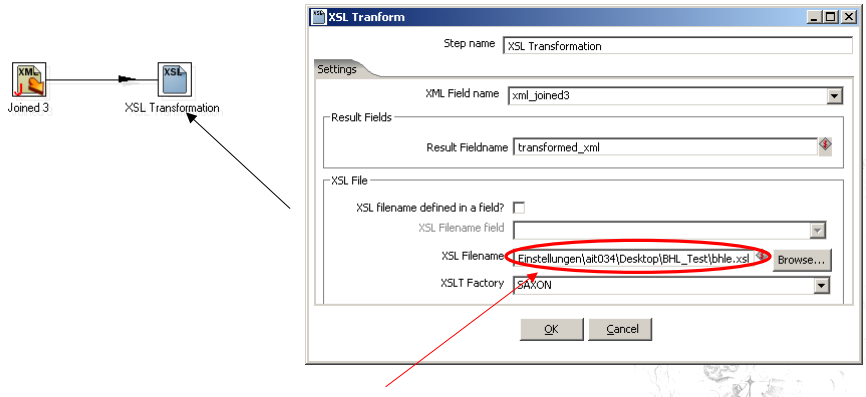
Joining XML Parts



These XMLs are merged into a single XML field.

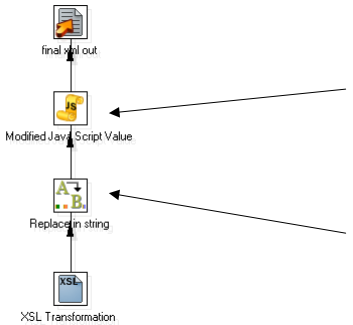


Applying a XSL Transformation



A XSL file can be specified to perform a transformation on the incoming XML field in order to adapt it to a specific XML schema, e.g. MODS.

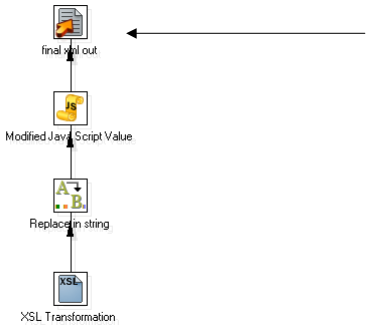
Finalizing the Transformation (1)



increments the loop variable for the evaluation in the parent job

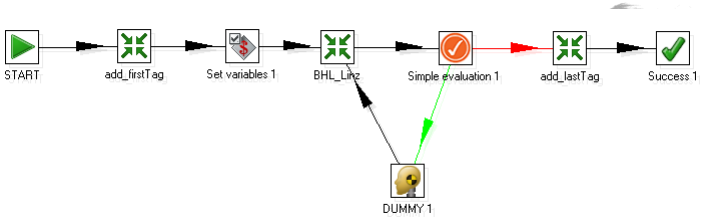
replaces some unwanted strings

Finalizing the Transformation (2)



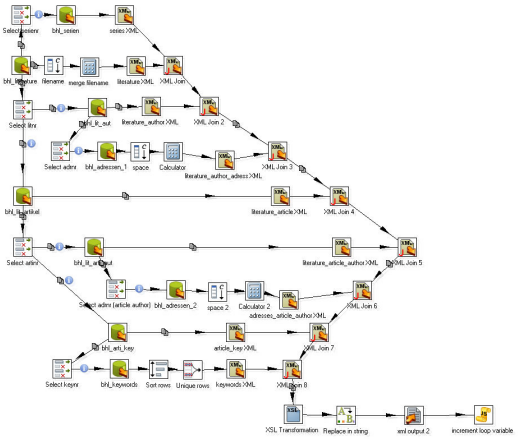
The text file output step is used to store the transformed information in a XML file.

Landesmuseum Oberösterreich (LANDOE) - Job



This job is equal to the one used for the MOBOT data.

Landesmuseum Oberösterreich (LANDOE) - Transformation



Transforming Data from XML Input

```
- <collection>
  - <record>
      <leader>01275nam a2200397 i 4500</leader>
      <controlfield tag="001">CSIC000378334</controlfield>
      <controlfield tag="005">20000501070000.0</controlfield>
      <controlfield tag="008">930623s1993 esp 000 0 eng
        d</controlfield>
    - <datafield tag="019" ind1="" ind2="">
        <subfield code="a">M-93-11292</subfield>
      </datafield>
    - <datafield tag="020" ind1="" ind2="">
        <subfield code="a">84-00-07353-3</subfield>
      </datafield>
    - <datafield tag="040" ind1="" ind2="">
        <subfield code="a">CIRBIC</subfield>
        <subfield code="b">spa</subfield>
        <subfield code="c">VL</subfield>
      </datafield>
    - <datafield tag="080" ind1="2" ind2="4">
        <subfield code="a">681.3</subfield>
      </datafield>
    - <datafield tag="080" ind1="2" ind2="4">
        <subfield code="a">57.087</subfield>
      </datafield>
    - <datafield tag="080" ind1="2" ind2="4">
        <subfield code="a">519.22</subfield>
      </datafield>
    - <datafield tag="100" ind1="1" ind2="">
        <subfield code="a">Marcus, Leslie F.</subfield>
      </datafield>
```

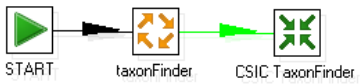
The input XML file in MARC schema is transformed into MODS schema.

(Note: Namespaces need to be removed before processing a file)

This example is a record from the *Consejo Superior de Investigaciones Científicas* (CSIC)

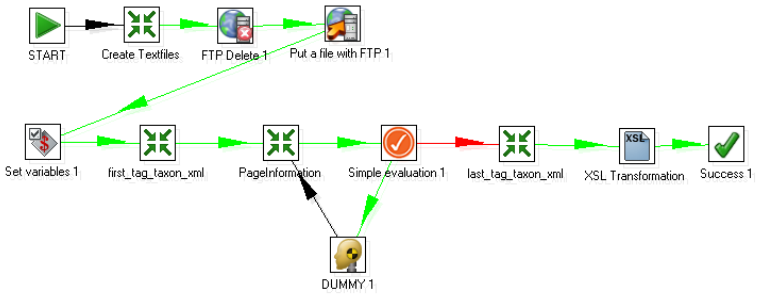
Consejo Superior de Investigaciones Científicas (CSIC)

Creates CSIC record and merges it with the Information from taxonFinder

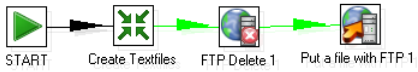


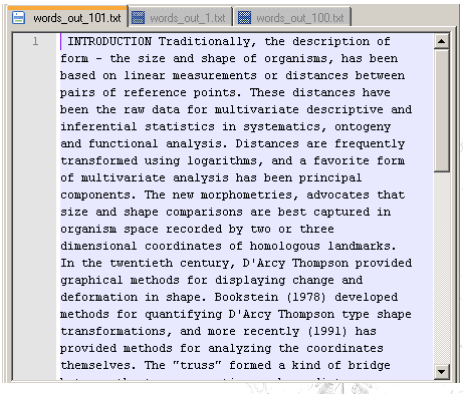
Uses Full-Text of the book to retrieve Expression from taxonFinder

Job: taxonFinder



Create Files and Transfer via FTP

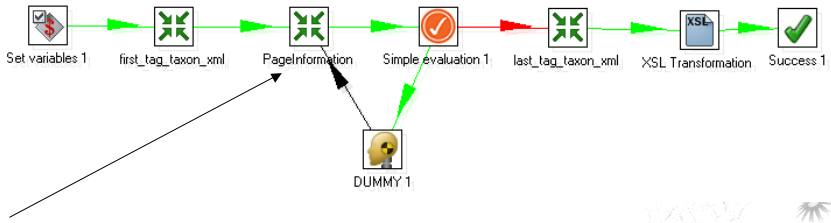




Text files are created for each page, containing all its words space separated.

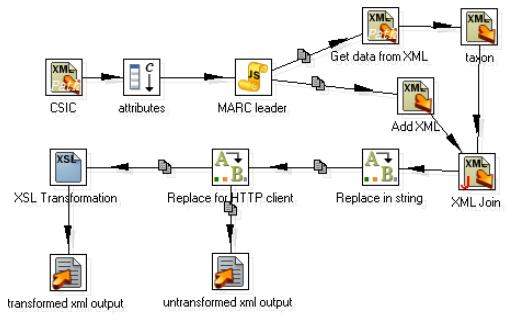
These files are then transfered via FTP in order to pass URL to the taxonFinder service.

Create XML containing taxonFinder words (1)

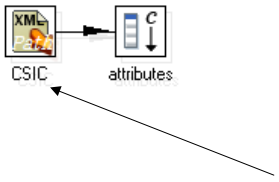


In the loop taxonFinder is called for every single page and the words occuring are saved with the appropriate link to the page in the Internet Archive.

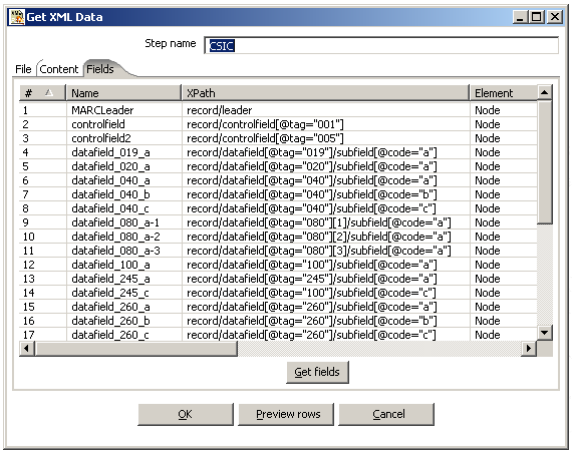
Merging the record with taxonFinder Information (1)



Getting Data from a XML Input File



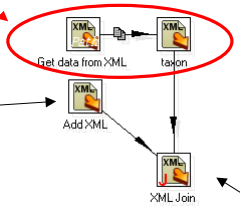
Fields can be specified using XPath statements. These fields can then be transformed as before.



#	Name	XPath	Element
1	MARCLeader	record/leader	Node
2	controlfield	record/controlfield[@tag="001"]	Node
3	controlfield2	record/controlfield[@tag="005"]	Node
4	datafield_019_a	record/datafield[@tag="019"]/subfield[@code="a"]	Node
5	datafield_020_a	record/datafield[@tag="020"]/subfield[@code="a"]	Node
6	datafield_040_a	record/datafield[@tag="040"]/subfield[@code="a"]	Node
7	datafield_040_b	record/datafield[@tag="040"]/subfield[@code="b"]	Node
8	datafield_040_c	record/datafield[@tag="040"]/subfield[@code="c"]	Node
9	datafield_080_a-1	record/datafield[@tag="080"][1]/subfield[@code="a"]	Node
10	datafield_080_a-2	record/datafield[@tag="080"][2]/subfield[@code="a"]	Node
11	datafield_080_a-3	record/datafield[@tag="080"][3]/subfield[@code="a"]	Node
12	datafield_100_a	record/datafield[@tag="100"]/subfield[@code="a"]	Node
13	datafield_245_a	record/datafield[@tag="245"]/subfield[@code="a"]	Node
14	datafield_245_c	record/datafield[@tag="100"]/subfield[@code="c"]	Node
15	datafield_260_a	record/datafield[@tag="260"]/subfield[@code="a"]	Node
16	datafield_260_b	record/datafield[@tag="260"]/subfield[@code="b"]	Node
17	datafield_260_c	record/datafield[@tag="260"]/subfield[@code="c"]	Node

Merging the record with taxonFinder Information (2)

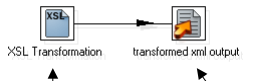
Reading XML file created with taxonFinder



CSIC record XML

Integrating taxonFinder expressions in the final XML.

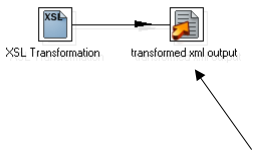
XML Output in MODS Schema (1)



Adapting to MODS schema

```
<?xml version="1.0" encoding="UTF-8" ?>
- <mods:mods xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:prov="http://www.openarchives.org/OAI/2.0/provenance"
    xmlns:mods="http://www.loc.gov/mods/v3"
    xsi:schemaLocation="http://www.loc.gov/standards/mods/v3/mods-
    3-3.xsd">
  - <mods:titleInfo>
      <mods:title>Contributions to morphometrics /</mods:title>
    </mods:titleInfo>
  - <mods:name>
      <mods:namePart>Marcus, Leslie F.</mods:namePart>
    </mods:name>
    <mods:typeOfResource>text</mods:typeOfResource>
  - <mods:originInfo>
      <mods:issuance>monographic</mods:issuance>
    </mods:originInfo>
  - <mods:originInfo>
    - <mods:place>
        <mods:placeTerm type="text">Madrid :</mods:placeTerm>
      </mods:place>
    </mods:originInfo>
  - <mods:originInfo>
      <mods:publisher>Museo Nacional de Ciencias Naturales,
        C.S.I.C.,</mods:publisher>
    </mods:originInfo>
```

XML Output in MODS Schema (2)



Useful Links/References

- Official Website of the Business Intelligence Suite: http://www.pentaho.org
- Website of Pentaho Data Integration: http://kettle.pentaho.org
- A beginners tutorial: http://etl-tools.info/en/pentaho/kettle-spoon.htm
- Pentaho Data Integration User Guide: http://wiki.pentaho.com/display/EAI/Spoon+User+Guide