# Data Warehousing

**Daniel Lemire**

# LECTURE 1
# A practical introduction to ETL

## 1.1. Introduction

We can do ETL without any specialized software. We can also use proprietary software from large vendors such as Oracle. Or, yet, we can use open source specialized software. There are at least two good such products: Talend and Pentaho Data Integration (PKI). We shall survey PDI.

## 1.2. You need Java

PDF is written in Java. You must have Java 5 or bettter.

## 1.3. Setting it up

- Grab PDI version 3.2.0 from source forge[1]. The file is a little over 60 MB.
- Uncompress the archive on your hard drive.

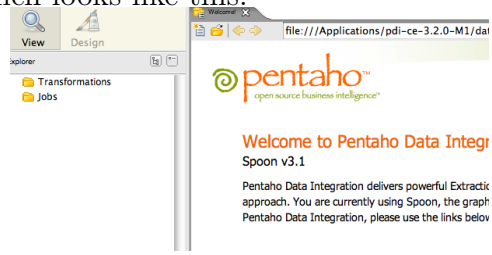
## 1.4. Launch Spoon!

Spoon is the graphical interface of PDI. Execute the script Spoon.bat under windows or `sh spoon.sh` under Unix or OSX.

[1] `http://downloads.sourceforge.net/pentaho/pdi-ce-3.2.0-M1.zip?modtime=1233246593&big_mirror=0`

You should get a screen looking like this:


Select the button `No Repository`. You should now see a screen which looks like this:




## 1.5. Some practice

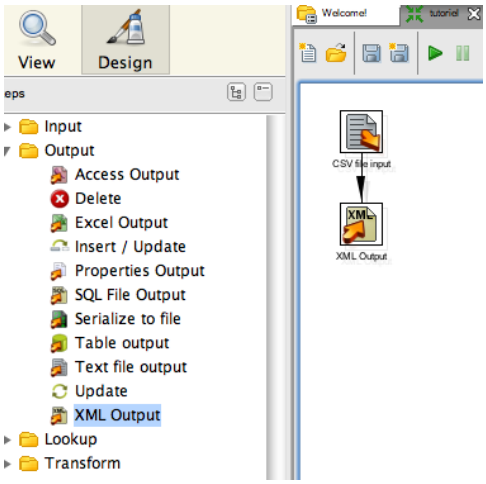
PDI allows you to take data from differente sources and transform them. As a first example, we will transfrom a CSV file into an XML file.

Begin by saving the following CSV data on your disk:

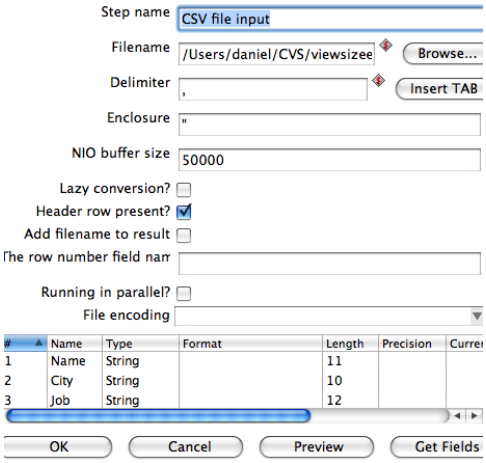
```
# this is just for unit testing
John, Montreal, Salesman
Kamel, Lyon, Researcher
Nathalie, Montreal, Translator
Bush, Washington, President
Tim Bray, Vancouver, Technologist
Daniel, Montreal, Professor
Peter, Montreal, Professor
Jack, Montreal, Professor
Nathalie, Montreal, Professor
John, Montreal, Salesman
```

```
Kamel, Lyon, Researcher
Nathalie, Montreal, Translator
Bush, Washington, President
Tim Bray, Vancouver, Technologist
Daniel, Montreal, Professor
Peter, Montreal, Professor
Jack, London, Professor
Nathalie, Montreal, Professor
John, Montreal, Salesman
Kamel, Lyon, Researcher
Nathalie, Montreal, Translator
Bush, Washington, President
Tim Bray, Vancouver, Technologist
Daniel, Montreal, Professor
Peter, Montreal, Professor
Jack, Montreal, Professor
Nathalie, Montreal, Professor
John, Dubai, Salesman
Kamel, Lyon, Researcher
Nathalie, Montreal, Translator
Bush, Washington, President
Tim Brayish, Vancouver, Technologist
Daniella, Montreal, Professor
Peterish, Montreal, Professor
Jackie, Montreal, Professor
Nathalie, Montreal, Professor
John, Montreal, Salesman
Kamel, Lyon, Researcher
Nathalie, Montreal, Translator
Bush, Washington, President
Tim Bray, Vancouver, Technologist
Daniel, Montreal, Professor
Peter, Montreal, Professor
Jack, Montreal, Professor
Nathalie, Montreal, Professor
```

Click on `New` then `Transformation`. On the left side of the screen, you will see an Input directory: drop the CVS file input icon on the workspace. Selec the CVS file input icon while pressing the Shift key and drag it to the XML Output icon. You should then see a window which looks like this:

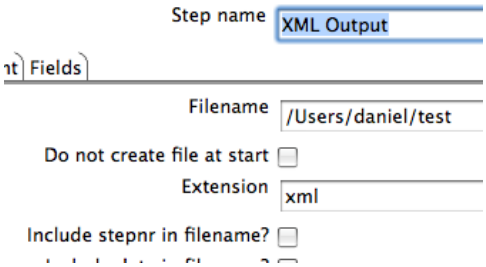



Double-click on the CSV File Input icon. Select the CSV file on your disk and uncheck Lazy Conversion, as in this picture:




Press Ok to continue.

Double-click on the XML Output icon. Enter a path such as C:\test (Windows) or /Users/myname/test (under MacOS) in the filename field. You should then have a window looking like this:




Press ok and continue.

Finally, back in the main window, hit Transformation, then Run, puis appuyez sur Launch.

If you did everything right, you should find a file test.xml on your disk with the following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<Rows>
 <Row><Name>John</Name> <City>Montreal</City> <Job>Salesman</Job> </Row>
 <Row><Name>Kamel</Name> <City>Lyon</City> <Job>Researcher</Job> </Row>
 <Row><Name>Nathalie</Name> <City>Montreal</City> <Job>Translator</Job> </Row>
 <Row><Name>Bush</Name> <City>Washington</City> <Job>President</Job> </Row>
 <Row><Name>Tim Bray</Name> <City>Vancouver</City> <Job>Technologist</Job> </Row>
(...)
```

Congratulations! You just transformed a CSV file into XML!

## 1.6. A reference before we continue

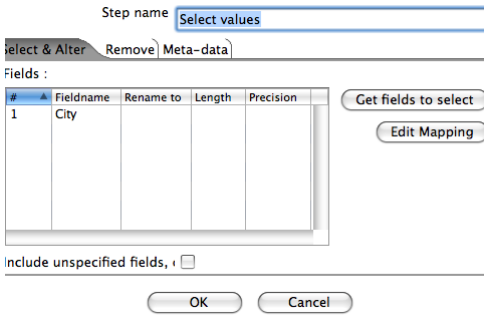
Pentaho makes available a tutorial at `http://wiki.pentaho.com/display/EAI/Pentaho+Data+Integration+(Kettle)+Tutorial`.

## 1.7. Ok, let's try something harder

Select XML Output, right-click and select detach.

We will now filter data to keep only the city names. Grab the Select values from the file Transform and drop it on your workspace. While pressing Shift, drag CSV file input on this new icon. Repeat the process from Select values to XML Output. You should have a workspace which looks like this:

Double-click on Select values and select only the City field as in this picture:

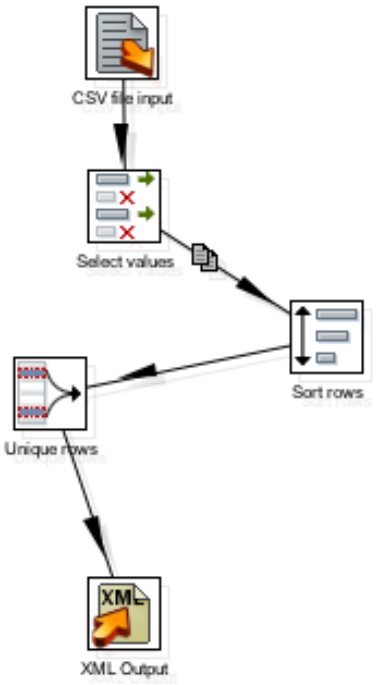



Hit Ok. Back in the main window do Transformation then run.

If everything went well, you should have an XML file which looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<Rows>
 <Row><City>Montreal</City> </Row>
 <Row><City>Lyon</City> </Row>
 <Row><City>Montreal</City> </Row>
 <Row><City>Washington</City> </Row>
 <Row><City>Vancouver</City> </Row>
 <Row><City>Montreal</City> </Row>
 <Row><City>Montreal</City> </Row>
 <Row><City>Montreal</City> </Row>
 <Row><City>Montreal</City> </Row>
 <Row><City>Montreal</City> </Row>
 <Row><City>Lyon</City> </Row>
 (...)
```

Ok. Let's go a little be further. Imagine that you want to remove duplicates. Grab the Sort rows and Unique rows icons and link them as in this picture:




Double-click on the Sort rows icon and select City. Hit Ok and restart the transform.= You should now have an XML file which looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<Rows>
 <Row><City>Washington</City> </Row>
 <Row><City>Vancouver</City> </Row>
 <Row><City>Montreal</City> </Row>
 <Row><City>Lyon</City> </Row>
 <Row><City>London</City> </Row>
 <Row><City>Dubai</City> </Row>
</Rows>
```

Joins

PDI can also link data by computing a join. Create two CSV files. The first one has an identifier and link to a city. The second one links the identifier to a name.

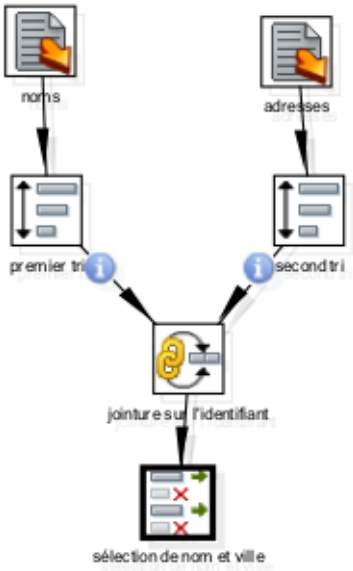
```
identifier,city
123, Montreal
44,Drummondville
```

```
identifier,name
123,Jean
44,Pierre
```

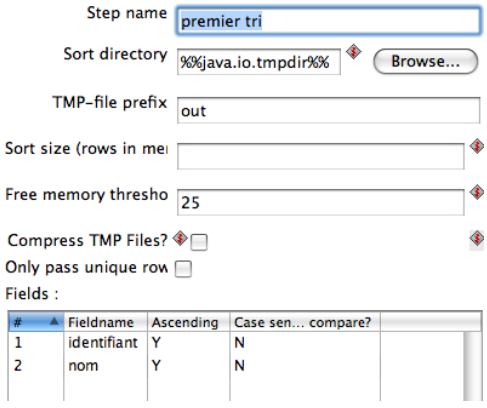
We wish to get the following result:

```
name,city
Jean, Montreal
Pierre,Drummondville
```

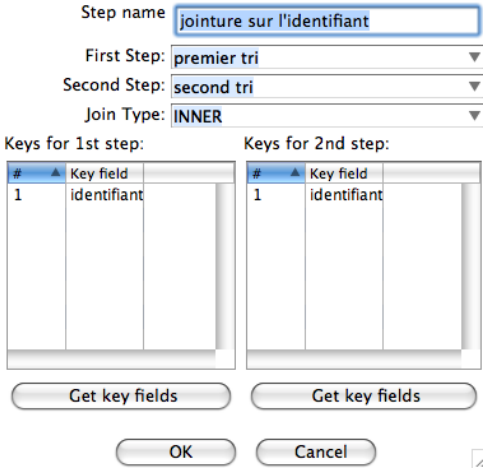
To get it, we use the Merge Join tools (from the Joins file) and Select Values (from the Transform file):



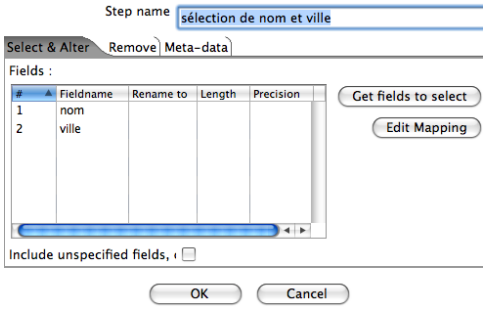

The tools Sort Rows (form the Transform file) are used to prepare for a Merge Join by sorting the data on the identifier:




The Merge Join is done on the identifier:




The last step (Select Values) selects the two fields we wish to keep (name and city):



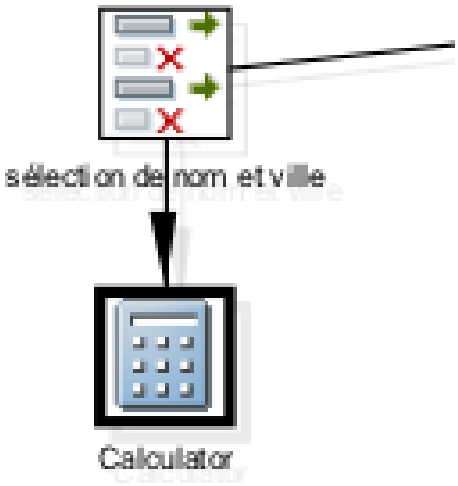

We could now expert the result to an XML file or a database.

## 1.8. Deriving new fields

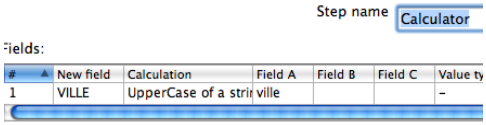
We can create new fields with the Calculator field. Imagine that you want to add the CITY field to store the city names in upper case:

```
name,city, CITY
Jean, Montreal,MONTREAL
Pierre,Drummondville,DRUMMONDVILLE
```

Just add the tool Calculator (from the Transform file):




and configure it with the right settings:




Several other functions are available, including the sum or product of existing fields and so on.

## 1.9. Group By

Another frequent operation in data warehousing is the group by. Suppose that you want to transform the original file

```
City,Sales
Montreal,10.00
Quebec, 20.00
Montreal, 10.00
```

in a file containing the total sales:

```
City,Total Sales
Montreal,20.00
Quebec, 20.00
```

The Group By tool (from the Transform file) will do it for you. Just remember to sort the data on the group (here city) before using it:


You then have to specify the group (city) and define a new field which will be the total of sales:

The fields that make up the group:

| # | Group field |
|---|---|
| 1 | City |

Aggregates :

| # | Name | Subject | Type |
|---|---|---|---|
| 1 | TotalSales | Sales | Sum |

## 1.10. Your homework assignment

I give you three CSV files:

```
Produit, Categorie
Bouzmagic, Maison
Plouzmagic Plus Plus, Pelouse
Bouzmagic Plus, Maison
Plouzmagic Plus, Pelouse
Plouzmagic, Pelouse
```

```
Produit, Prix
Bouzmagic, 2.44
Plouzmagic Plus Plus, 1.99
Bouzmagic Plus, 19.00
Plouzmagic Plus, 22.00
Plouzmagic, 12.59
```

```
Produit, Ventes
Bouzmagic, 432
Plouzmagic Plus Plus, 43211
Bouzmagic Plus, 800
Plouzmagic Plus, 809
Plouzmagic, 4000
```

- You must join the three files on the field `Produit`.
- You must compute for each `Produit` the total of sales, that is the product of the fields `Prix` and `Ventes`. Output the answer as an XML file.
- Regroup the sales per Categorie: compute the total of sales for each Categorie. The answer must be an XML file.

Make sur to include a screen shot (PNG, GIF or JPG) showing you solution from within the PDI tool, the two XML files and a complete discussion (in English) of your solution.