

# ITR Utility

© 2007, Ashish Banerjee ([www.AshishBanerjee.com](http://www.AshishBanerjee.com))

Free & open source Java utility to generate XML file from CSV file mapped by a property file.

## Introduction

ITR Utility is a general purpose, simple utility that lets you convert CSV (Comma separated Values) into an XML file, without using a transformation language like XSLT.

This utility was created on request of the Indian Chartered Accountants' (CA) community, to enable them to generate XML files for filing income tax returns (ITR). The CSV files may be generated from a spreadsheet or a database. The ITR formats will change in near future. However, this utility empowers the CA to create their own XML files from sample files, without needing to know the XSLT programming complexities.

ITR Utility is simple and easy to learn and targeted for non programmers.

## Download & Installation

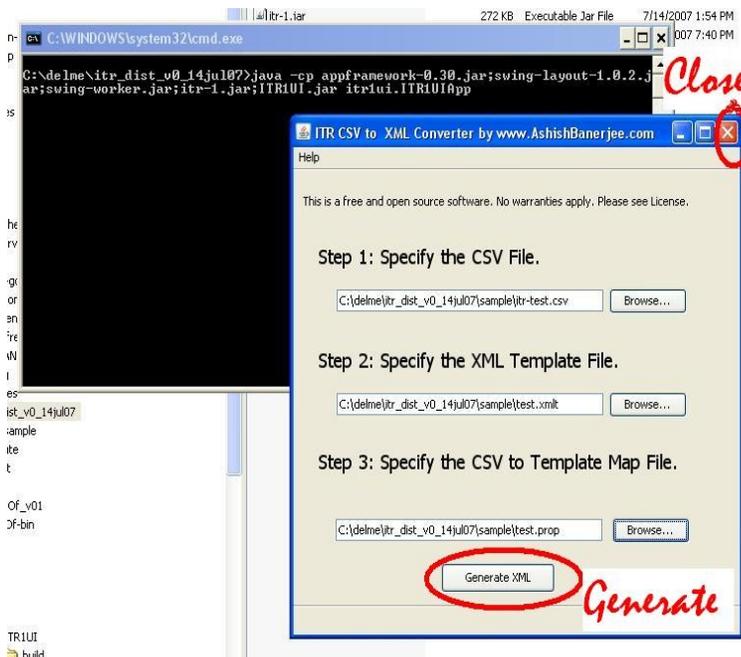
Download the utility from [http://www.ashishbanerjee.com/itr\\_dist\\_v0\\_14jul07.zip](http://www.ashishbanerjee.com/itr_dist_v0_14jul07.zip) and unzip it into a directory. You can use the Windows XP / Vista built-in tool for this purpose.

If you do not have Java runtime (Java 1.6) download and install it from <http://www.java.com/en/download/index.jsp>

Execute the utility by double clicking on **run.bat** file.

## Getting Started

On executing the run.bat you should see a screen like this below:



The ITR Utility needs three input files. See the sample directory contents.

The screenshot illustrates the workflow of the ITR Utility. It shows five windows:

- Spreadsheet:** An OpenOffice Calc window showing a table with columns A-N and rows 1-5. The data is as follows:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Name	Ravinder												
2	Surname	Gupta												
3	Net income	700000												
4	Agricultural Income	10000000												
5	Tax	100000												
- Template file:** A Notepad window showing an XML template with variables: `<?xml version="1.0" encoding="UTF-8"?>`, `<Test xmlns="http://AshishBanerjee.com/test" >`, `<FirstName>${FirstName}</FirstName>`, `<Lastname>${Lastname}</Lastname>`, `<Income>${Income}</Income>`, `<AgriIncome>${AgriIncome}</AgriIncome>`, `<Tax>${Tax}</Tax>`, and `</Test>`.
- Property (template map):** A Notepad window showing a text file with mappings: `FirstName=B1`, `LastName=B2`, `Income=B3`, `AgriIncome=B4`, and `Tax=B5`.
- CSV file:** A Notepad window showing a CSV file with the following content:

```
"Name","Ravinder"
"Surname","Gupta"
"Net income",700000
"Agricultural Income",10000000
"Tax",100000
```
- Generated XML:** A Notepad window showing the resulting XML file with the variables replaced by their values: `<?xml version="1.0" encoding="UTF-8"?>`, `<Test xmlns="http://AshishBanerjee.com/test" >`, `<FirstName>Ravinder</FirstName>`, `<Lastname>Gupta</Lastname>`, `<Income>700000</Income>`, `<AgriIncome>10000000</AgriIncome>`, `<Tax>100000</Tax>`, and `</Test>`.

As we know, the use case is to create a XML file from a given spreadsheet. In order to generate the output XML, you will need to create a template file. The easiest way to create a template file is to take an existing sample XML file, and mark it up with the variables.

The template variable have the format:

```
 ${X}
 ${FirstName}
```

Note that the variables are case sensitive and are decorated by a dollar sign and enclosed within two curly brackets. See the **sample/test.xmlt** file. You can open it in notepad application or any text editor.

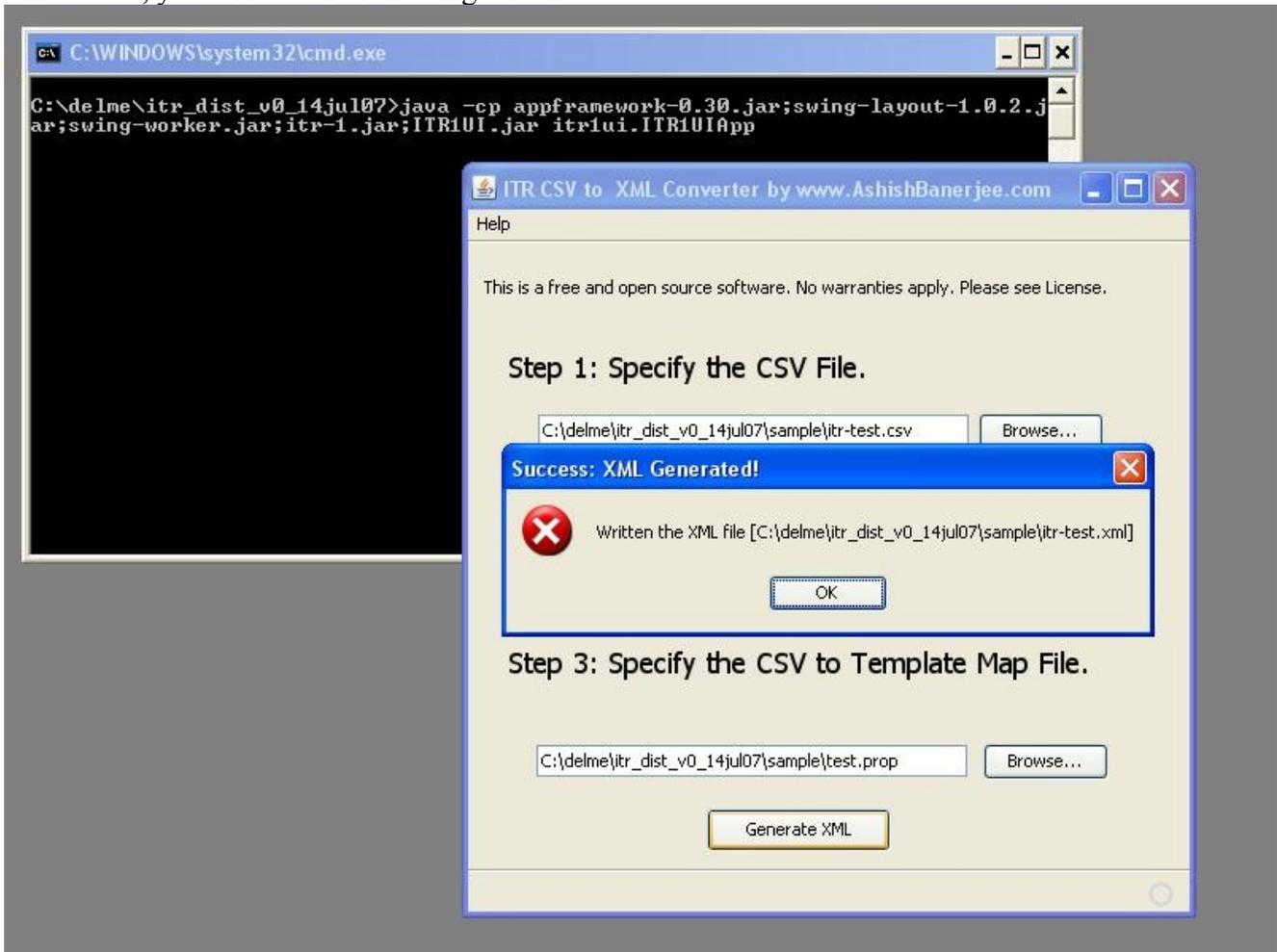
The Property (or Template Map) file is also a text file. Use a text editor to edit it. This file maps the spreadsheet cells to the variable names, without the decorative dollar or the curly brackets. It has the following simple format:

```
FirstName=B1
```

The case sensitive variable name, followed by an equals sign then the cell number. The first cell is numbered as A1. See the file **sample/test.prop**

Finally you need to save the spreadsheet as CSV file. Then provide these file names to the ITR Utility. Press the generate button. A XML file will be created, it will be named as the CSV file, but having

XML extension. For example if the CSV file is **itr-test.csv** then the generated xml will be **itr-test.xml**  
On success, you will see the following screen.



## **Files**

readme.txt : Basic instructions  
run.bat : execution batch file  
jar files: Java runtimes  
sample directory: Sample files  
src directory: two Netbeans 6 projects with sources.

## **Disclaimer:**

It is a free and open source product, with absolutely no warranties. No support is provided.