Summer Fellowship Report

On

Mapping Metadata to Doublincore in Dspace

Submitted by

Kavitha.K

Under the guidance of

Dr. Manju Naika
Chief Library Officer
Central Library, IIT Bombay

June 29, 2020

Acknowledgement

I am grateful to my **parents and teachers** for their everlasting inspiration and support. The internship opportunity I had with IIT, Bombay was a great chance for learning and professional development. I express my deepest gratitude and special thanks to the **Dr. Manju Naika**, chief library officer who in spite of being extraordinarily busy with his duties, took time out to hear, guide and keep me on the correct path during the fellowship. It is my radiant sentiment to place on record my best regards, deepest sense of gratitude to **Mrs.Samrudhi** Hawaldar , Sr. Library Information Asst., **Mr.Bhagwan Nagapure**, Sr.Library Information Asst., **Mr.Pravin Ghorpade** Sr. Project Technical Assistant for taking part in useful decision & giving necessary advices and guidance and arranged all facilities according to my convenient. I perceive as this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives. Hope to continue cooperation with all of you in the future; I perceive as this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives. Hope to continue cooperation with all of you in the future, I feel great pleasure to thank all my **friends** for giving me support to complete my project in a great success manner. I also thank each and every **person** who has helped me directly or indirectly for completing of this project work.

Contents:

Chapter-1:

1.1Introduction:

Dspace is an open source repository software package typically used for creating open access repositories for scholarly and/or published digital content. While Dspace shares some feature overlap with content management systems and document management systems, the Dspace repository software serves a specific need as a digital archives system, focused on the long-term storage, access and preservation of digital content.

1.2 Features of Dspace:

Some most important features of Dspace are as follows.

- Free open source software
- Completely customizable to fit user needs
- Manage and preserve all format of digital content (PDF, Word, JPEG, MPEG, TIFF files)
- Apache SOLR based search for metadata and full text contents
- UTF-8 Support
- Interface available in 22 languages
- Granular group based access control, allowing setting permissions down to the level of individual files
- Optimized for Google Scholar indexing
- Integration with BASE, CORE, Open AIRE, Unpaywall and WorldCat

1.3 Technology:

Dspace is constructed with Java web applications, many programs, and an associated metadata store. The web applications provide interfaces for administration, deposit, ingest, search, and access. The asset store is maintained on a file system or similar storage system. The metadata, including access and configuration information, is stored in a relational database and supports the use of PostgreSQL and Oracle database. Dspace holdings are made available primarily via a web interface. More recent versions of Dspace also support faceted search and browse functionality using Apache Solr.

1.4 Objective of the Project:

The objective of the project is to make bulk import of data from one database to another database. In this project data from source database is done manually that is preparing the CSV format of required data. Then the processed CSV is imported into another database using bulk import method.

Chapter-2:

2.1 Software Requirements for Dspace:

- Jdk
- Apache-ant
- Apache-maven
- Apache-tomcat
- PostgreSQL
- OS: Windows, linux, unix, solaris,

2.2 Software under my Project:

- Base OS : Windows 10 -64 bit
- Oracle VM Virtual Box
- Ubuntu 16.04 -64 bit
- Xampp server

2.3 Target of the Project:

Retrieving data from one database to another database. In this case, we are having one source database and destination database. The project work is to migrate data from one database to another database successfully. The source database is MySQL and the destination database is PostgreSQL. In this project, data migration is done using the Dspace SAF builder tool. The tool by itself needs some pre-required softwares. They are JDK and Apache-maven. The tool supports in all types of Operating System. In this project Linux, operating system is used.

Chapter - 3:

3.1 Dspace Metadata Schema:

Metadata has been a buzzword in the digital library community for several years. As systems and schemas are being developed, several categories of metadata have been defined and named. Although the concepts behind the different categories of metadata are fairly consistent, there is some variance in how different groups (and individuals) categorize types of metadata, and there is often overlap between the different categories. For the purposes of Dspace I have chosen to define the following categories of metadata:

Descriptive Metadata provides identification and a description of an object. It is used primarily for search and retrieval purposes, providing information about the contents of an object. It provides fields for table-driven searching, as well as a description of the intellectual contents and a physical description of the object. Typical descriptive metadata includes information about the object's source, creation, and content, as well as subject classification and identifying tags. This is the type of metadata that has been traditionally provided by libraries in their catalogs, and by other agencies associated with the distribution of information items such as publishers and government agencies.

Structural Metadata provides information about the relationships between different parts of an object. It binds together components of complex information objects.

Administrative Metadata provides information used in managing and administering information resources within a system or a federation of systems. Typical examples of administrative data are information about rights and reproduction, legal requirements, version control, access restrictions, and statistical and audit trails.

Preservation Metadata provides information about the physical specification of an object's creation, its format and condition, hardware and software requirements to render it, it's transformation into other formats (change history or "provenance") and it's authenticity (fixity). The purpose of preservation metadata is to help future generations interpret and recreate the information objects.

3.2 Existing Metadata Schema vs. a Unique Dspace Schema

There are several advantages to using standard descriptive metadata schemas. The use of ratified standards will facilitate interoperability between Dspace and other repositories and systems. Crosswalks are being created between established metadata schemas to accommodate merging of different types of data collections and to allow global searches of more than one system through gateways and protocols. It is also possible that Dspace metadata will be exported to other systems. Even within MIT Libraries, there is already a crosswalk between the MARC communications format and the Dublin Core schema. Using standard schemas will allow Dspace to participate in these efforts to share content with other systems and federations of systems. Another advantage to using established metadata schemas is that we can take advantage of the years of deliberations, thought and work that have gone into the process of creating, using and standardizing these schemas.

3.3 DSpace Content

Interdisciplinary:

In deciding upon schemas for use with Dspace it is important to consider the type of content Dspace can expect to receive and process, both now and in the future. Since Dspace is an institutional repository, it will have to accommodate the many disciplines representing all the schools, departments, labs and centers at MIT. In addition, Dspace has committed itself to handling many types of digital formats, including text, data, image audio and video. In such a heterogeneous environment, it is unlikely that one particular descriptive metadata schema will fit all Dspace content. It is also impossible to forecast exactly which types of content will become prominent in the future. Communities that collect or produce different types of information objects have used different metadata schemas to serve their needs.

As we have approached potential contributors to Dspace we have noted the different types of metadata schemas used by different communities, and what data formats have prompted them to use them. We have also noted where metadata records already exist for specific collections. So far we have come across the Dublin Core schema for text files, the FGDC schema for geospatial data files, MARC records for scanned objects already in the MIT Libraries catalog, and two standards for digital images, DCMI.

Batch Submissions:

Several of our early adopter communities will be submitting batches of electronic items converted from print to electronic form. The first substantial batch of Dspace content will be a collection of pdf images of MIT Press out-of-print books. Subsequent batches of submissions to Dspace will consist of collections of scanned pdf images of print technical reports and working paper series from various labs, centers and schools on campus. Most of the items in these collections will have already been cataloged by MIT librarians, and therefore we will have access to MARC format metadata records for them.

Individual Submissions:

It is expected that the bulk of regular and ongoing submissions to DSpace will be done on an individual basis, by the creator of the content object or by an administrative staff member of the community. During the submission process, each submitter will be expected to fill out a metadata form providing at least minimal information describing the submitted document. Any schema Dspace adopts will have to be simple enough for non-catalogers to use, yet also be able to capture the information elements needed to provide adequate searching capabilities. Metadata input templates will have to be easy to use and not too lengthy, so that potential contributors to Dspace will not be put off by the submission process.

Dublin core Metadata Schema:

The Dublin Core metadata schema (DC) was developed especially for electronic information objects to provide a simple metadata schema that would be flexible enough to accommodate a variety of formats. It consists of a simple "unqualified" set of fifteen elements, and an optional set of qualifiers that expand some of the original 15 elements. Accompanied by well-designed templates for metadata input, the Dublin Core schema has the potential to elicit enough information to provide ample searching and identification capabilities for Dspace. Its disadvantage is that it may not provide the fine-grained targeted fielded search capabilities that some communities producing specialized data objects would require. It also contains some elements that are not contained in the MARC record, such as the "format" element and the "rights" element.

Chapter – 4:

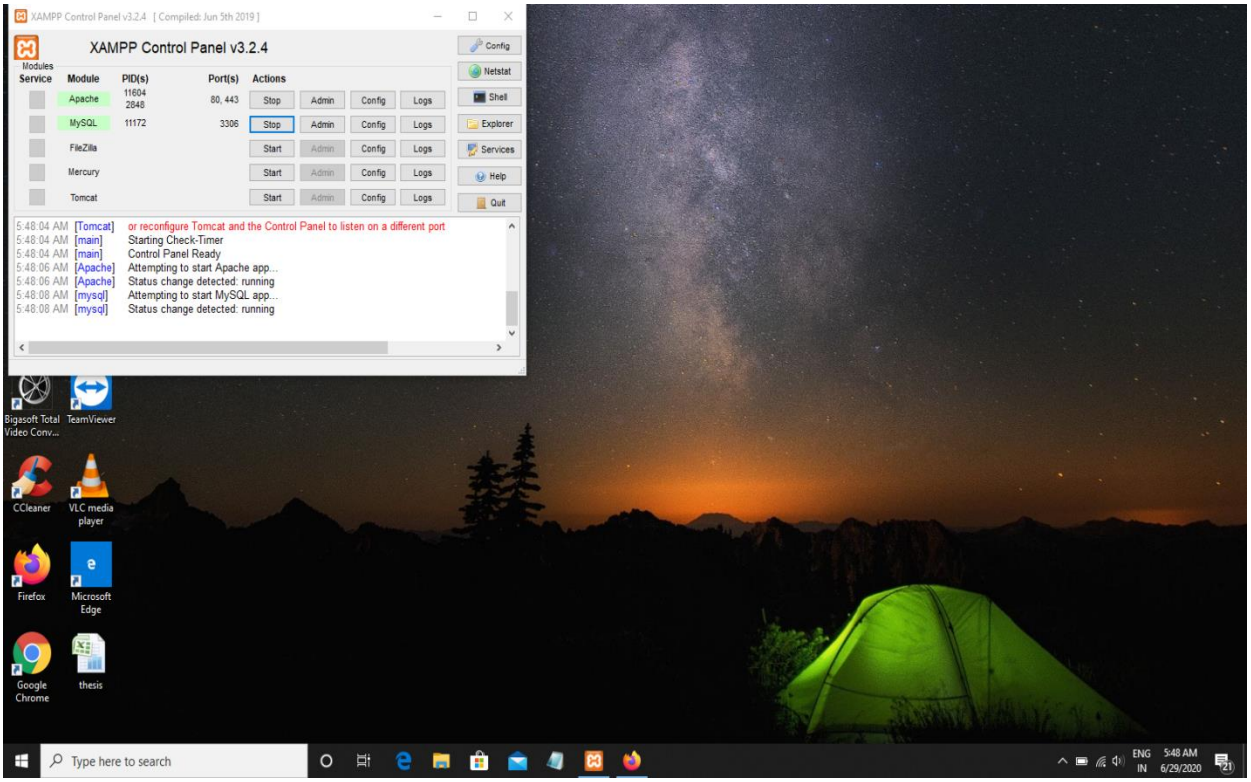
4.1 Retrieving Data:

First of all we are having the data in MySql Database. In this project Xampp server is used as a source database. For testing process the data are retrieved from the ETD server through Google Drive and then imported into the MySql database which acts as a client in the Xampp server. But the actual source and destination are the ETD server and the Dspace. Retrieving data can be done through various procedures. In this project to retrieve data from MySql, we can download the database as CSV file. The screenshots are given below.

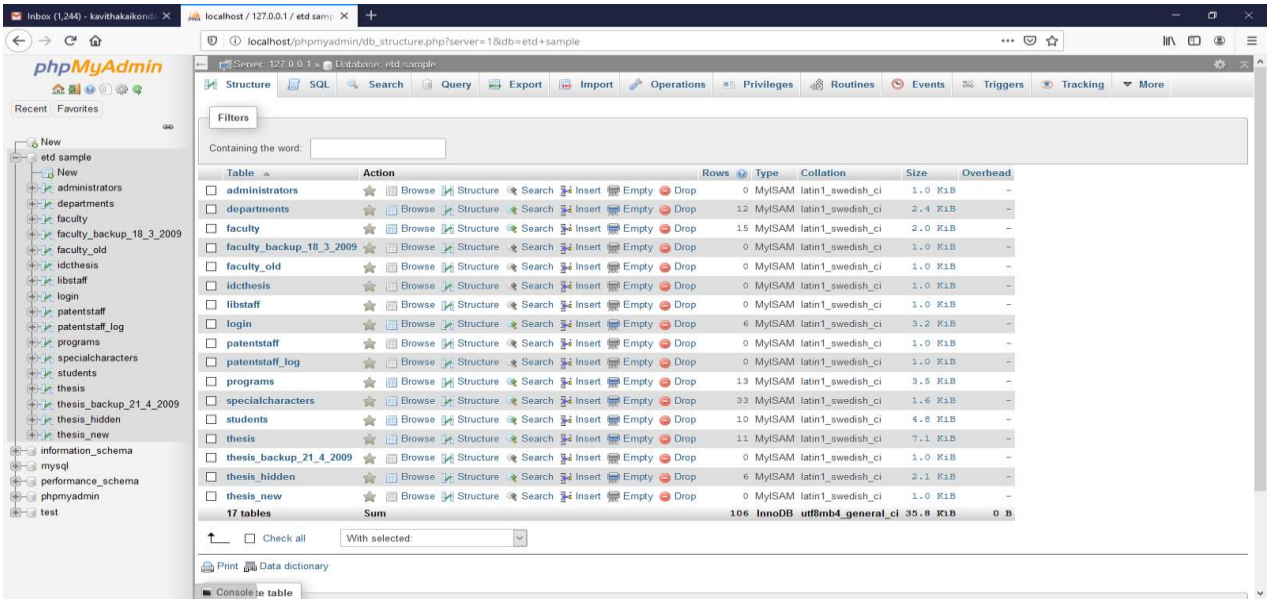
A Comma Separated Values (CSV) file is a plain text file that contains a list of data. These files are often used for exchanging data between different applications. For example, databases and contact managers often support CSV files.

These files may sometimes be called Character Separated Values or Comma Delimited files. They mostly use the comma character to separate (or delimit) data, but sometimes use other characters, like semicolons. The idea is that you can export complex data from one application to a CSV file, and then import the data in that CSV file into another application.

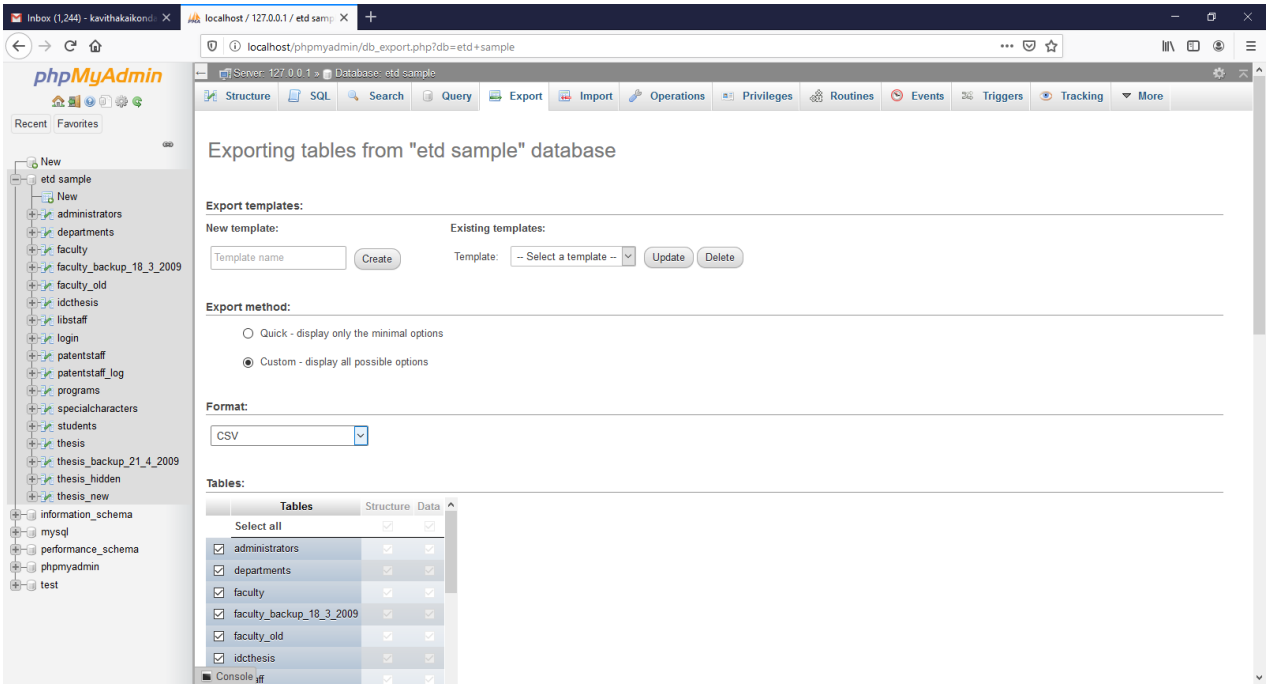
Xampp Server Control Panel for MySQL:



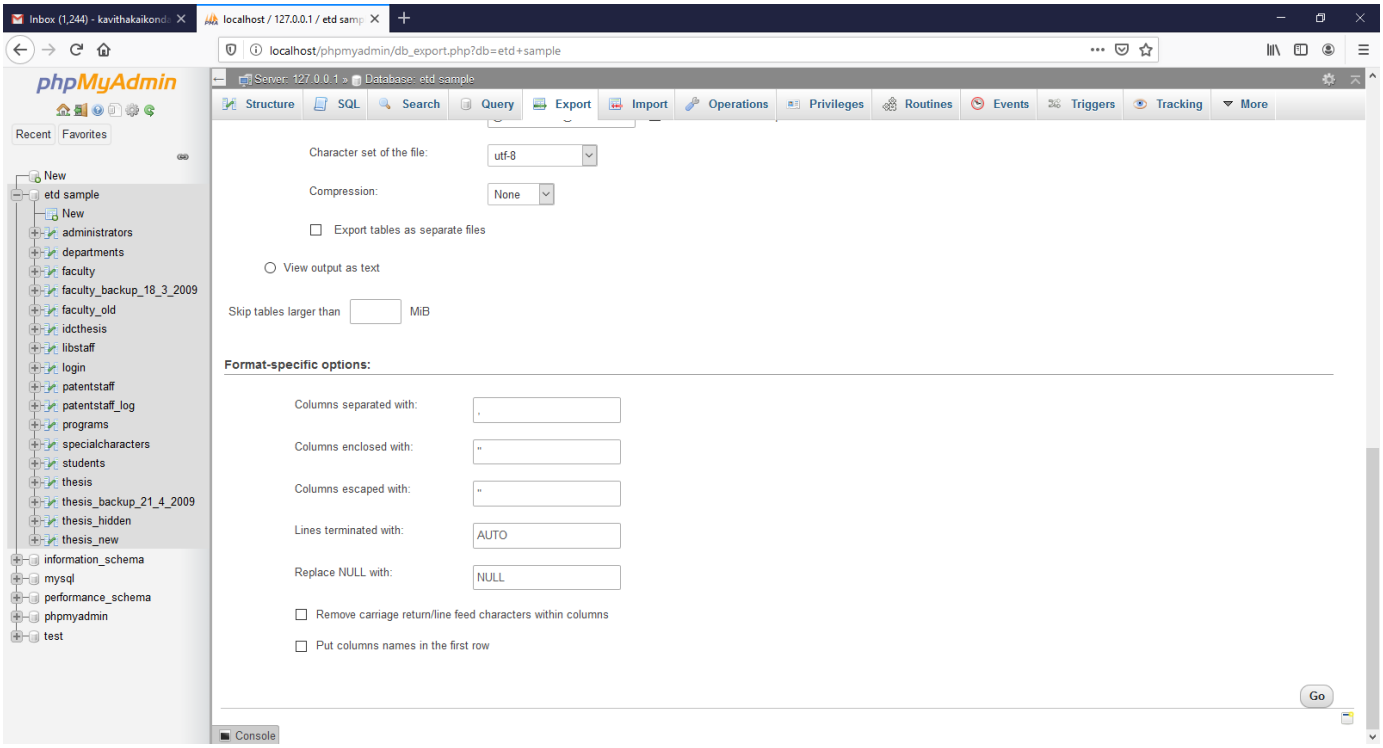
Structure of the MySQL Database:



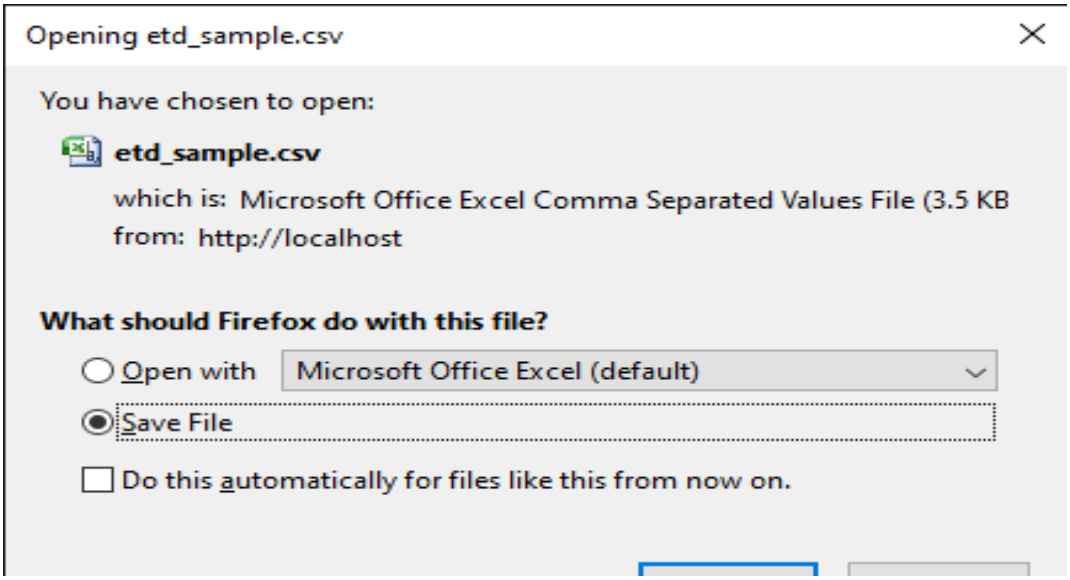
Exporting database into CSV:



Downloading Database in to CSV :



Downloading database into CSV:



4.2 Editing CSV:

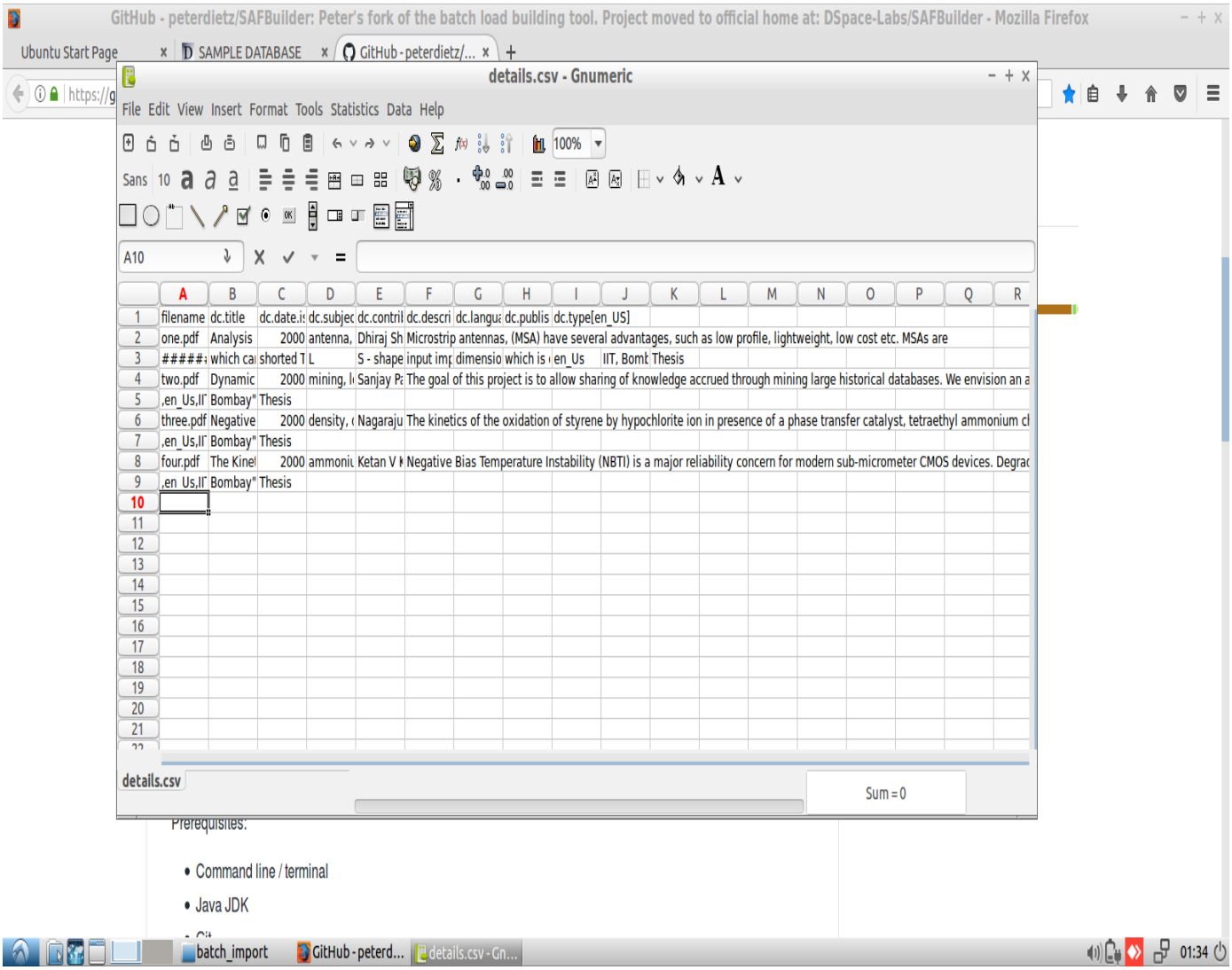
By default, Microsoft Excel may not correctly open the CSV in Unicode/UTF-8 encoding. This means that special characters may be improperly displayed and also can be "corrupted" during re-import of the CSV.

You need to tell Excel this CSV is Unicode, by importing it as follows. (*Please note these instructions are valid for MS Office 2007 and 2013. Other Office versions may vary*)

- First, open Excel (with an empty sheet/workbook open)
- Select "Data" tab
- Click "From Text" button (in the "External Data" section)
- Select your CSV file
- Wizard Step 1
    - Choose "Delimited" option
    - Start import at row: 1
    - In the "File origin" select box, select "65001 : Unicode (UTF-8)"
        - NOTE: these encoding options are sorted alphabetically, so "Unicode (UTF-8)" appears near the bottom of the list.
    - Click Next
- Wizard Step 2
    - Select "Comma" as the only delimiter
    - Click Next

- Wizard Step 3
    - Select "Text" as the "Column data format" (*Unfortunately, this must be done for each column individually in Excel*)
        - At a minimum, you MUST ensure all date columns (e.g. dc.date.issued) are treated as "Text" so that Excel doesn't auto convert Dspace's YYYY-MM-DD format into MM/DD/YYYY
        - To avoid such auto conversion, it is safest to ensure each column is treated as "Text". Unfortunately, this means selecting each column one-by-one and choosing "Text" as the "Column data format".
    - Click Finish
- Choose whether to open CSV in the existing sheet or a new one

Screenshot of edited CSV is given below:



4.3 Installing pre required Software & Dspace SAF builder:

Dspace SAF Builder:

A tool that turns content files and a metadata spreadsheet into a Simple Archive Format package, which easily allows for batch import to Dspace, an Institutional Repository.

The input for a command-line batch ingest of materials to Dspace is well documented, and is called "Simple Archive Format", however there needs to be a tool that easily facilitates creating a Simple Archive Format package. The use case satisfied with the Simple Archive Format Packager is that someone has a spreadsheet filled with metadata as well as content files that are eventually destined for repository ingest.

Thus the input to the Simple Archive Format Packager is a spreadsheet (.csv) that has the following columns:

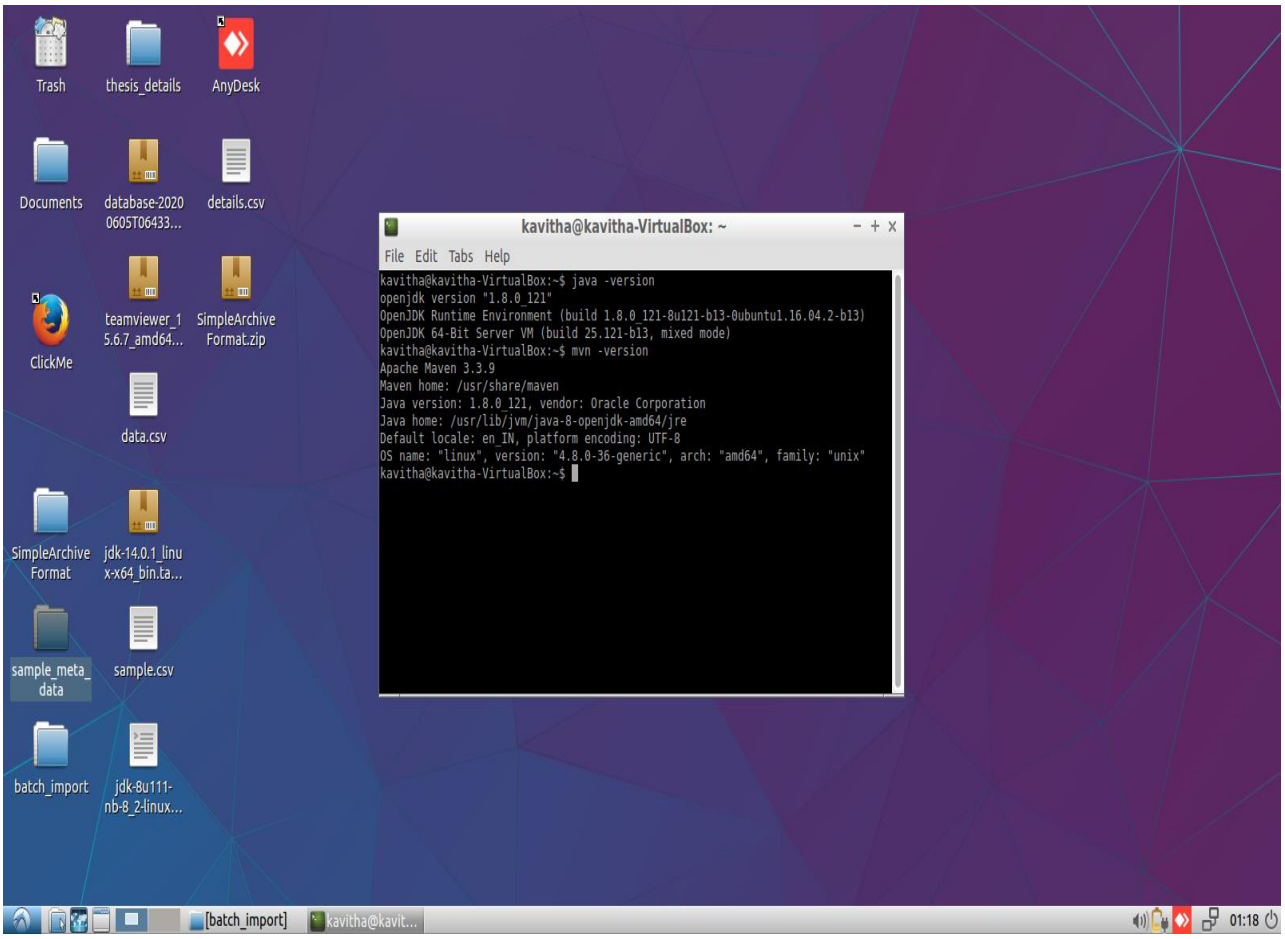
- **filename** of the content file(s)
- **namespace.element.qualifier** metadata for the item. Examples would be: dc.description or dc.contributor.author

Further, dates need to be in ISO-8601 format in order to be properly recognized. And for any column that has multiple values, you can separate each entry with a double-pipe "||". For example, for multiple files just set "filename" to "file1.pdf||file2.pdf||file3.pdf". Similarly, multiple "dc.subject" values can be separated by "||"

Prerequisites:

- Command line / terminal
- Java JDK
- Git
- Maven

The following screenshot shows that needed prerequisites are installed successfully:

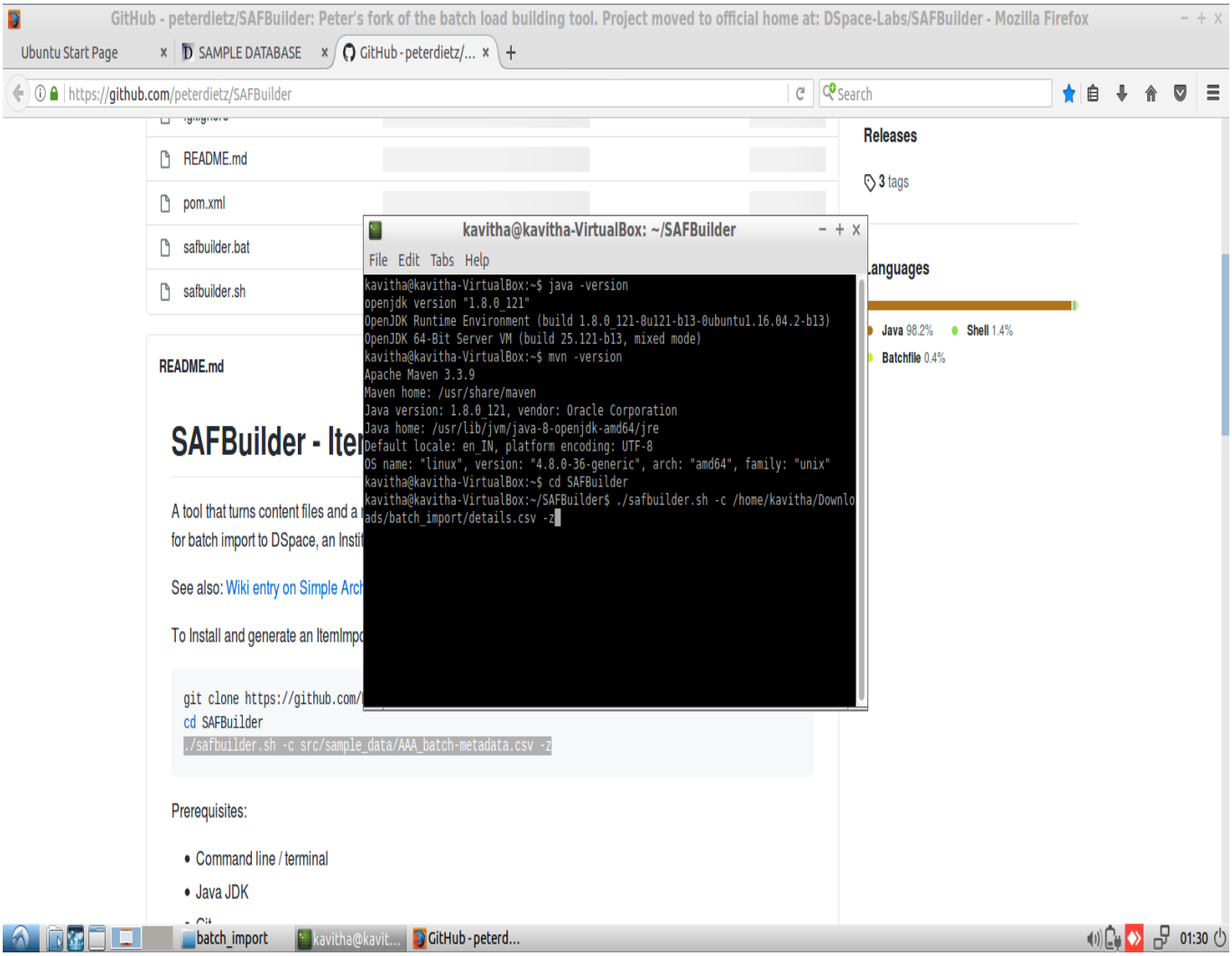


Installing Dspace SAF builder tool:

The Dspace SAF builder tool can be installed using the github cloning method or we can download the zip file of the tool. The tool gets installed successful, when the pre required softwares are installed successfully.

Github link: https://github.com/DSpace-Labs/SAFBuilder

The following screenshot shows that the Dspace SAF builder is installed successfully.

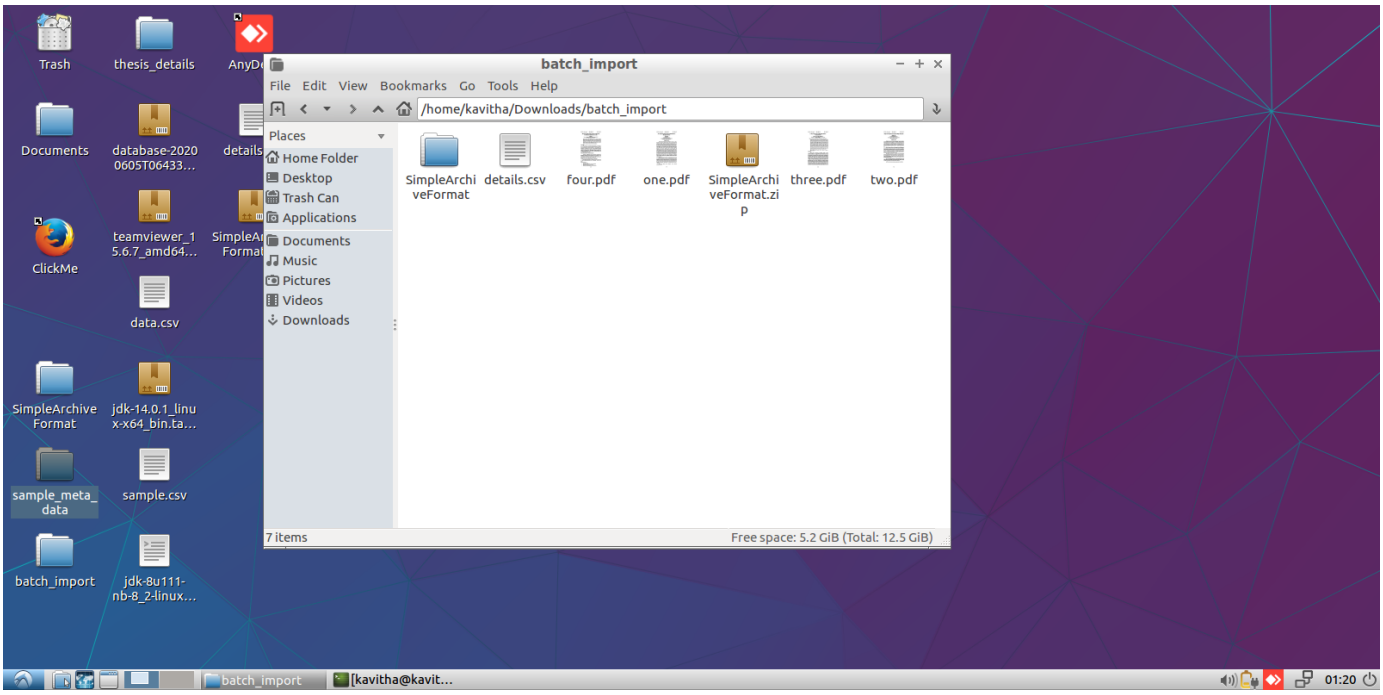


4.4 Processing the edited CSV in Dspace SAF builder tool

After this, now we have to process the edited CSV file using the Dspace SAF builder tool using commands in the terminal. Then it will provide the output as Zip file which we required. The output gives exact zip file as Simple Archive Package. Because in Dspace there is an option for uploading such Zip file.

For processing the CSV, we need to create a folder with CSV and its supporting documents like PDF, JPEG etc. It is shown in the below screenshot

Folder containing CSV file and its supporting files:



After all these preparation we need to pass the below command for processing:

```
cd SAFBuilder
```

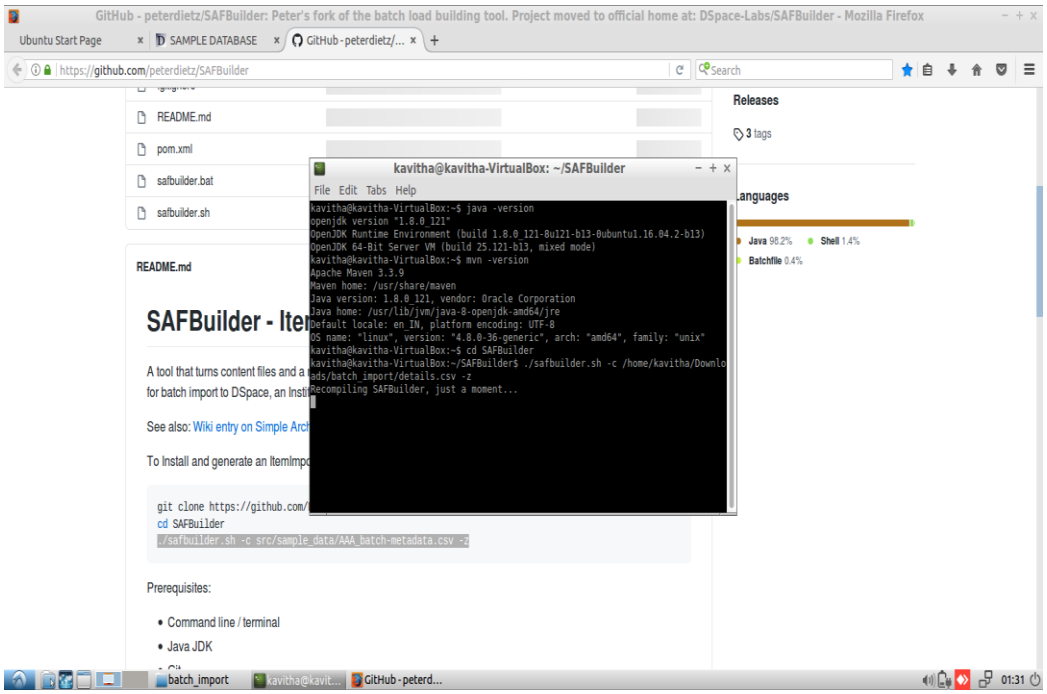
```
./safbuilder.sh -c /home/kavitha/Downloads/batch_import/details.csv -z
```

NOTE: Path of the folder should be given correctly, and then only we can get output in the same path.

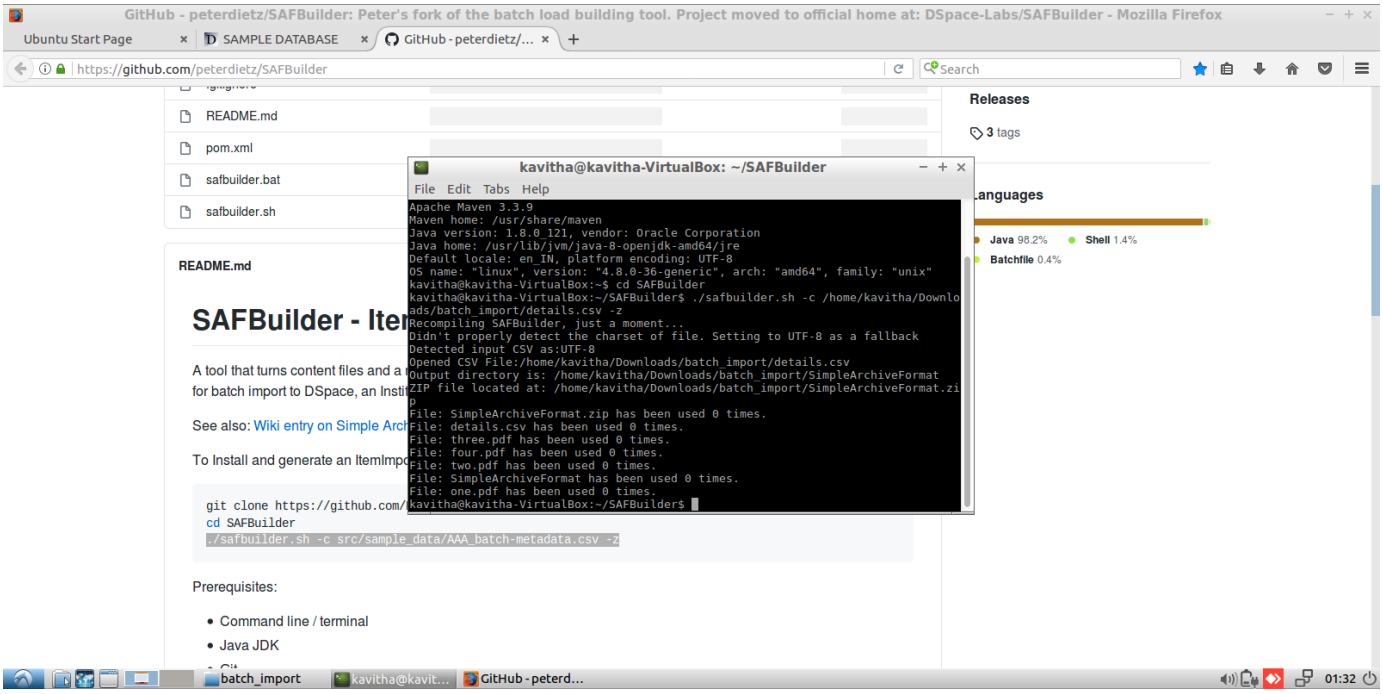
Finally, after passing the above commands we can get correct output in the exact path.

Screenshots of that process are given one by one.

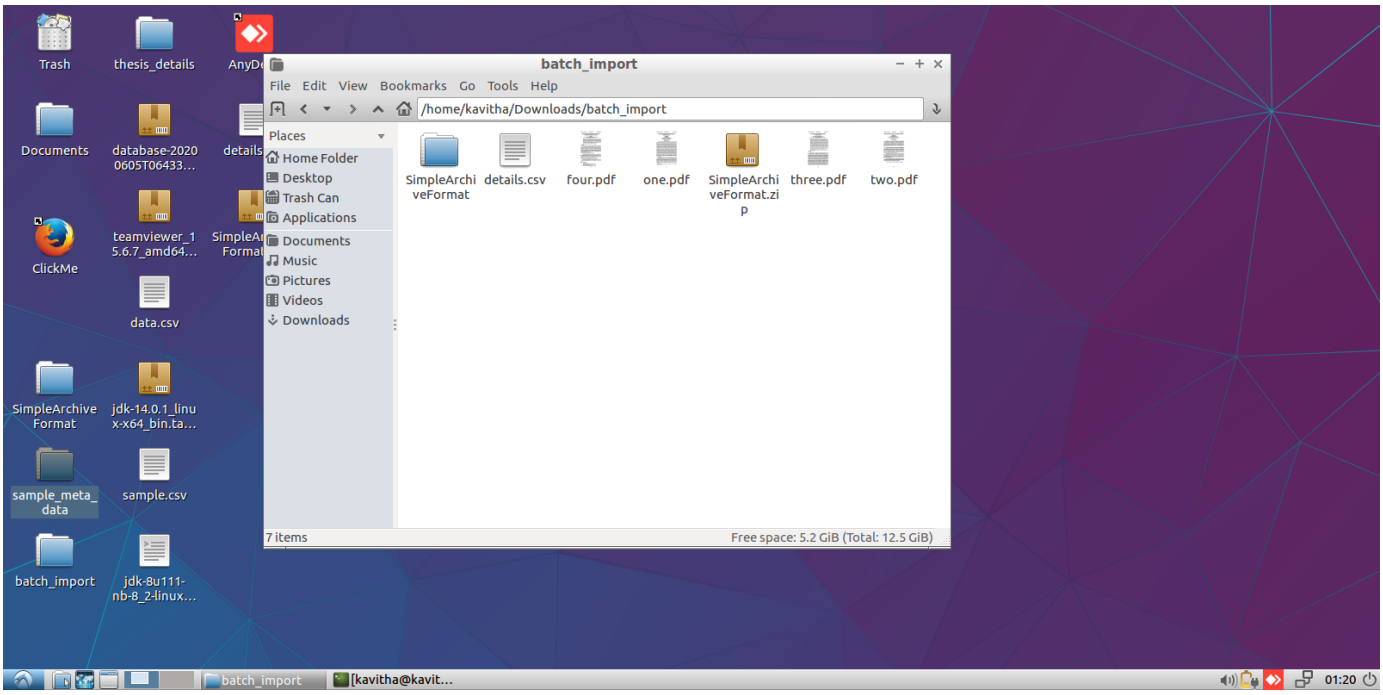
Processing the files:



Output in Command Line:



Output as Zip file in the path:



4.5 Uploading the result in Dspace:

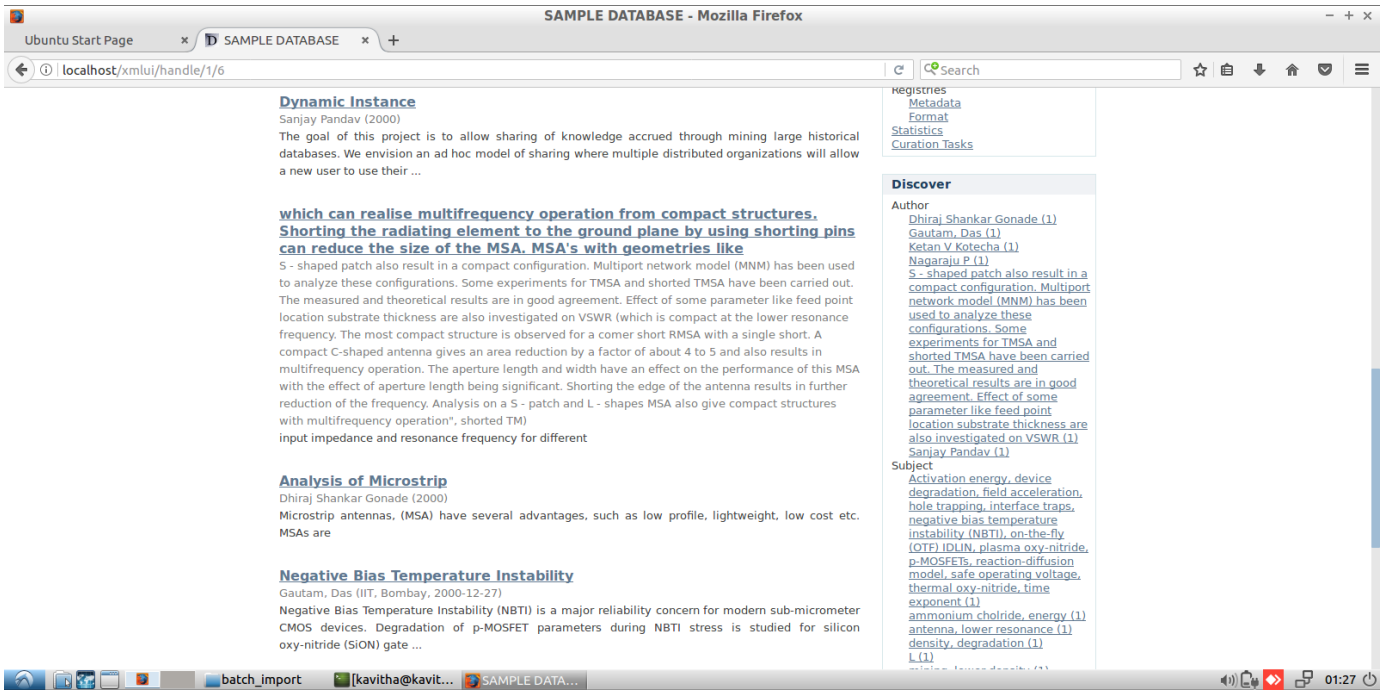
The batch import system for Dspace is a simple but primitive method for importing multiple items into a Dspace repository. Typically, a custom script will need to be created which migrates data from a source into this Dspace batch import format. Possible source formats are Excel spreadsheets, Access databases, other websites, or a flat file system. The custom script will produce a set of directories and files that correspond with the format.

The dublin_core.xml contains the descriptive metadata about the item. This well-formed XML file is simple a list of <dcvalue> elements, each with its Dublin Core element, qualifier and value. You will need to check with the repository administrator about what Dublin Core Element and Qualifiers are available, and which values should be placed in these fields.

Output:

```
SimpleArchiveFormat/
  item_000/
      dublin_core.xml          -- qualified Dublin Core metadata for metadata
fields belonging to the dc schema
      metadata_ [prefix].xml    -- metadata in another schema, the [prefix] is
the short name of the schema as registered with the metadata registry
      Contents                 -- text file containing one line per filename
      file_1.doc               -- files to be added as bitstreams to the item
      file_2.pdf
  item_001/
      dublin_core.xml
      contents
      file_1.png
  item_...
```

The below screenshot shows the uploaded result in the Dspace :



References:

https://wiki.lyrasis.org/display/DSPACE/Simple+Archive+Format+Packager

https://wiki.lyrasis.org/display/DSDOC5x/Batch+Metadata+Editing

https://github.com/DSpace-Labs/SAFBuilder

Websites worked:

Dspace URL: http://localhost/xmlui/