

EMERSE Virtual Machine (VM)

For additional help, or to provide feedback regarding this manual, please contact David Hanauer, at: hanauer@umich.edu

Additional information about EMERSE: <http://project-emerse.org>

Last updated March 15, 2017

INTRODUCTION

EMERSE (Electronic Medical Record Search Engine) is provided on a virtual machine (VM) as a way for potential users to test the system and even load their own data on it for searching. It should be noted at the outset that this virtual machine approach is intended primarily for light use as a demonstration and not for full production-level use in a large research or operational environment. For large-scale implementations EMERSE should be installed on dedicated servers and be maintained by a professional IT team, and all security and regulatory considerations must be addressed.

The following directions were created to help small groups pilot EMERSE on a VM, but the process for loading documents into EMERSE described below are not what we recommend for a full-scale installation of EMERSE. For more detailed technical directions on the full-scale implementation and integration, please see our technical manual.

This demo version of EMERSE on the VM uses an Oracle database that is an *Express Edition*, which means that it is free to distribute but limited in its power. It can use only one CPU and has a maximum allowed size of 4 GB. However, that is more than enough for this demonstration system, and it is also worth noting that the majority of the data used by EMERSE (i.e., the documents) are stored in Solr files and not in the Oracle database.

There are four main sections that follow:

GETTING STARTED: Describes basics for how to download both the Virtual Box application and the EMERSE virtual machine, and how to launch EMERSE, along with other details about exploring the system.

IMPORTING YOUR OWN DATA: Describes the process for importing your own data into the EMERSE VM for further testing or, in some cases, for small studies in which institutional IT support is not possible.

SECURITY CONSIDERATIONS: For those that do choose to import data, some security considerations should be addressed to ensure that the data remain safe.

MAKING EMERSE AVAILABLE ON THE HOST MACHINE: For those who want to run the VM essentially as a background server and want to be able to access the EMERSE application web page from their host Machine (e.g., Mac or Windows) rather than using the Linux interface within the VM.

GETTING STARTED

The directions in this section are provided to allow for basic downloading and launching of the EMERSE virtual machine, and to launch EMERSE itself for testing with the data that are pre-loaded into the system. PubMed abstracts are being used in place of clinical patient documents in this demonstration version.

Downloading and importing the VM

Note: This only needs to be done once. After it has been imported, you can start the VM over and over again by following the directions in the next section (Launching the EMERSE VM).

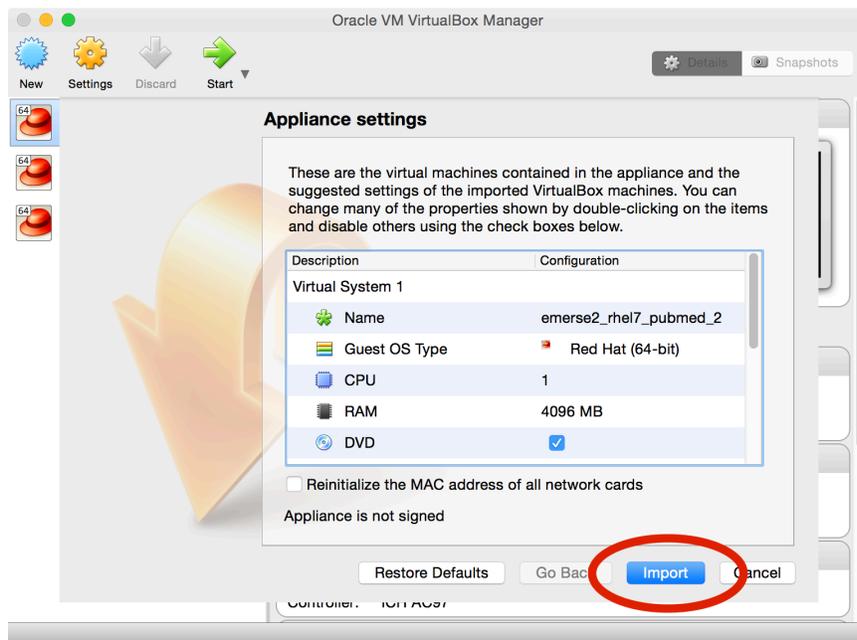
□ Download and install the VirtualBox application, if not done already. You can find it at:

<https://www.virtualbox.org>

□ Download the EMERSE virtual machine, which is packaged as an `.ova` file. Currently we are distributing this via a link that we will share with you, so feel free to contact us for the latest VM file. Contact David Hanauer at hanauer@umich.edu.

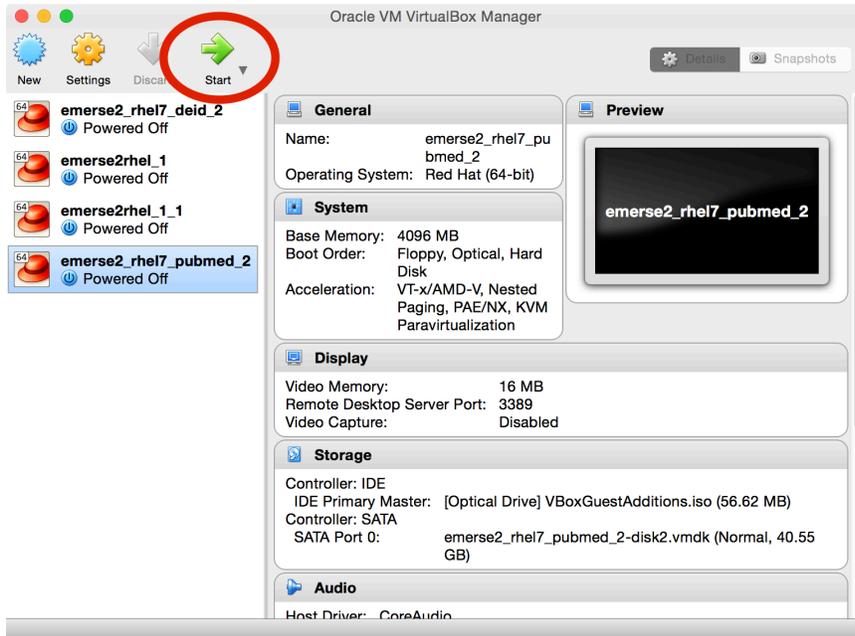
□ Double click on the `.ova` file. This will launch the VirtualBox application.

□ A window will open that is related to Appliance Settings. Click on the `Import` button.

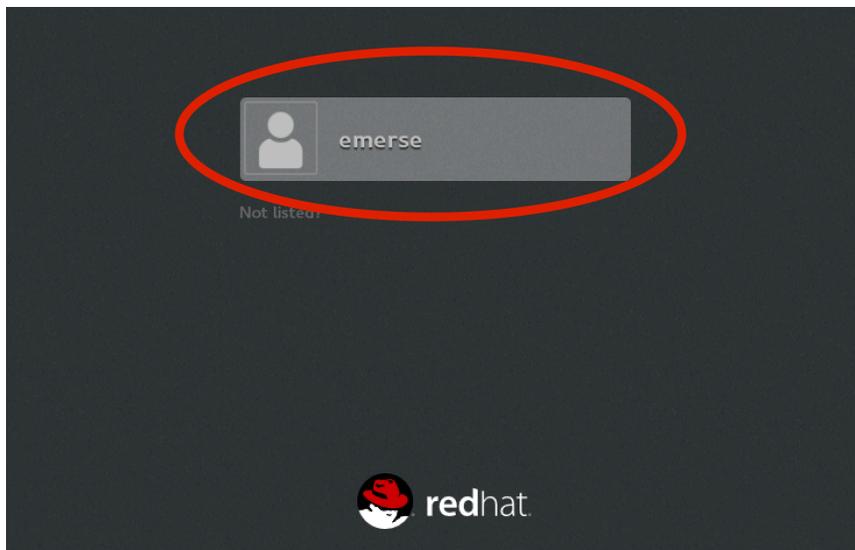


Launching the EMERSE VM

□ Once the VM is imported, make sure it is selected in the pane on the left in the main Virtual Box window, then press the green start arrow. This will launch the VM (a Linux operating system).



□ The default username for this Linux OS is `emerse` and the password is `demouser`. Once the OS loads you will see the login screen that says EMERSE. Click on where it says EMERSE, then enter the password on the next screen. This username and password is also the default for the EMERSE system on the VM (accessible via the web browser).



Launching EMERSE

Once the virtual machine (VM) is up and running, launching EMERSE should be simple. The servers that run EMERSE should already automatically start up, so all you need to do is launch the browser (preferably Chrome). This can be found on the Linux VM under `Applications → Internet → Chrome` or by double clicking on the Chrome icon on the Desktop. Then, simply go to the URL:

```
localhost:8090/emerse/login.html
```

```
username: emerse
```

```
password: demouser
```

This username and password combination are used throughout the VM, for the Linux OS, for the EMERSE application, for Oracle, etc.

Other useful things to explore on the VM

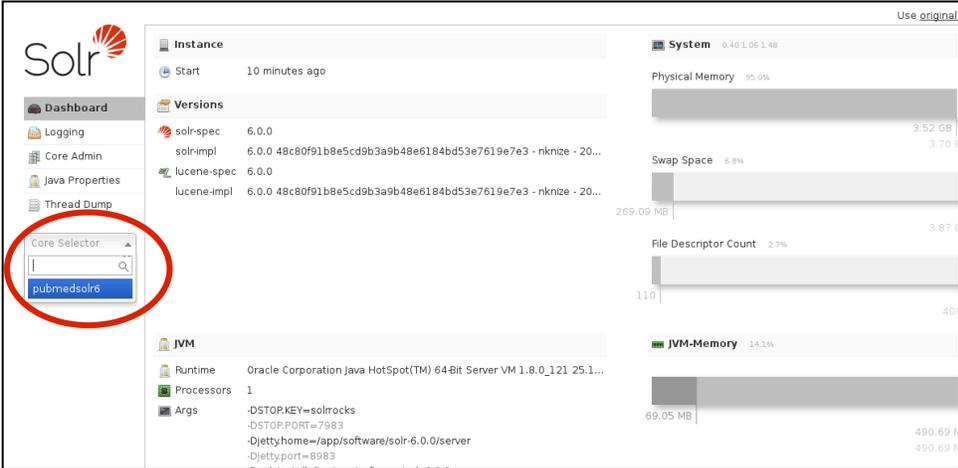
1. Besides the main EMERSE interface, there are other areas that may be worth exploring. For example, the EMERSE admin page (for adding more users) can be found at:

```
localhost:8090/emerse/admin.html
```

2. In addition, for those wishing to try out the native solr interface for issuing a query, visit this URL:

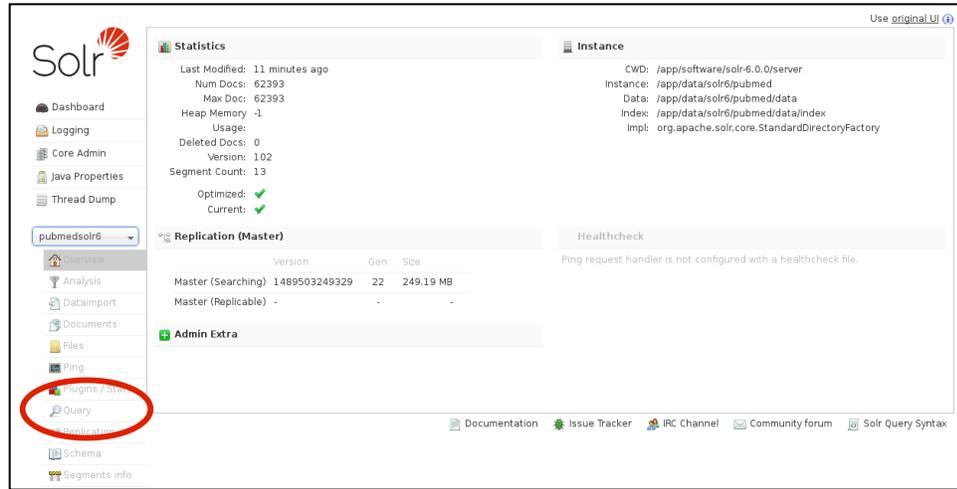
```
localhost:8983/solr/
```

To get to the proper area for running queries from this interface, click on the left pane where it says “Core Selector” and choose “pubmedsolr6”:

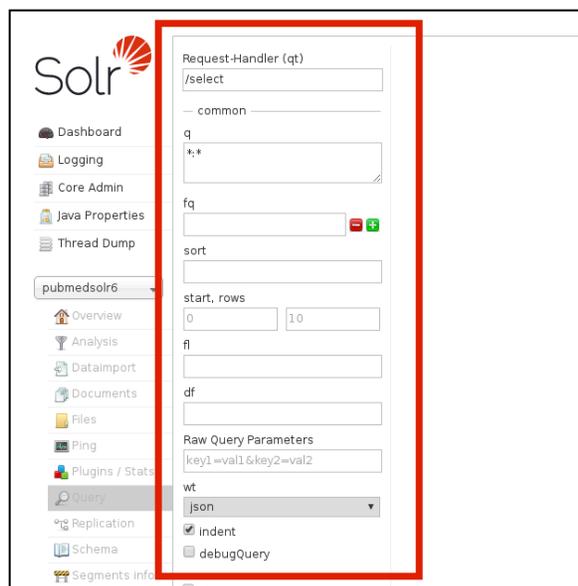


The screenshot displays the Solr Admin interface. On the left sidebar, the 'Core Selector' dropdown menu is highlighted with a red circle, and the option 'pubmedsolr6' is selected. The main content area shows the 'Instance' details, including the start time (10 minutes ago) and a list of versions (solr-spec, solr-impl, lucene-spec, lucene-impl). The right sidebar displays system metrics: Physical Memory (95.0%), Swap Space (6.8%), File Descriptor Count (2.7%), and JVM-Memory (14.1%).

Then, click where it says “Query”:



You will then be at the page where solr queries can be run, which can be useful for testing and troubleshooting queries:



3. To explore the Oracle database, in the Linux VM menu bar go to Applications → Developer → SQL Developer.

Then in the SQL Developer application click on Connections → Local XE as EMERSE.

username: emerse
password: demouser

IMPORTING YOUR OWN DATA

The following directions are only needed if there is a desire to load your own data into EMERSE for searching your own documents. It is not required for the basic demonstration version of EMERSE. It is also not recommended as an institutional solution for supporting multiple users. In other words, this approach could be used for individual projects where you are not able to obtain institutional support to install EMERSE centrally. In such a case, it is still important to consider the privacy and security implications of running your own VM to host data if the system were to contain patient data with protected health information. Please consult your local IT support group for advice on the security aspects.

For importing data, we assume that you will have a way to obtain all of your documents that need to be imported into EMERSE, with each document saved as a separate file.

The general steps for importing data into EMERSE are shown in the table below.

| Step | Description | Section(s) describing details in this document |
|------|---|---|
| 1 | Convert your documents for importing if they are not already in plain text or HTML formats. | Preparing documents for import: File Conversion |
| 2 | Creating metadata files that will define aspects of how the EMERSE system should display the documents, and define the patients with whom the documents will be associated. | Metadata files: Overview Metadata files: Document sources Metadata files: Document Source Metadata Metadata files: Document Metadata Metadata files: Patient Metadata |
| 3 | Creating a shared directory between your host computer and the guest virtual machine to copy the data from the host to the VM. This step, and the following step, is needed only if the data are created on the host machine. If the metadata files are created on the VM then there is no need to copy them. | Setting up a shared directory for copying data to the VM |
| 4 | Copy the files (metadata files and the actual documents for import) to the virtual machine. | Using the shared folder to copy the data files to the VM |
| 5 | Run the data loading scripts. | Using the scripts to load data into EMERSE |

Preparing documents for import: File Conversion (Mac directions only)

EMERSE uses Apache Solr for indexing documents. Solr can handle several file formats, but not all file formats are supported. If your documents are already plain text (.txt) or HTML (.html or .htm) then no additional conversion is required, and you can skip this section.

Files in other formats should be converted to either plain text or HTML. HTML is ideal for files with formatting (e.g., tables, bold text, etc) since HTML files can generally preserve that formatting whereas plain text files will not.

If file conversion is needed because the documents are not already in plain text (.txt) or HTML (.html or .htm) format, there are various ways to convert files, including several commercial options. The following directions were made for the Mac, using tools already provided with the Mac to perform file conversions. Other file conversions options are available for Windows machines. On Mac OS X there is a file converter built into the operating system. This can convert Microsoft Word documents (.doc or .docx) into HTML files which can then be imported into EMERSE. This file converter can also transform RTF documents into HTML. The following directions describe a simple approach for converting Word documents into HTML. Similar steps would be needed for converting RTF into HTML.

- Create a directory/folder on the Mac. It can have any name, but in this example you can name it `documentstoconvert` and it will be located on the Desktop.
- Copy all of the documents that need converting to this folder.
- Open a Terminal window. Navigate to Applications → Utilites → Terminal and double click the Terminal application to open it.
- At the Terminal command line, navigate to the proper directory, by typing:

```
cd ~/Desktop/documentstoconvert/
```

- Next, at the command line, convert the documents by typing:

```
textutil -convert html *.docx *.doc
```

Note that this will convert all .docx and .doc files into HTML format. To also include RTF files, simply add that part of the command, such as:

```
textutil -convert html *.docx *.doc *.rtf
```

The converted files should all have a .html extension. You can now either delete or remove the original files, or simply copy the new HTML files to a new folder to separate them out. Note that you can sort the folder by Kind to get all of the HTML files together in the list, which makes it easier to just select that file type.

Note: On the Mac it is also possible to convert PDF files, but the process is a bit more involved. This can be done using the Automator app, and developing a workflow. The PDF conversion component in the Automator app can be found under the PDFs → Extract PDF Text

workflow. One can use this Automator workflow save the files in RTF format, and then follow the directions about using the `textutil` app to convert RTF to HTML.

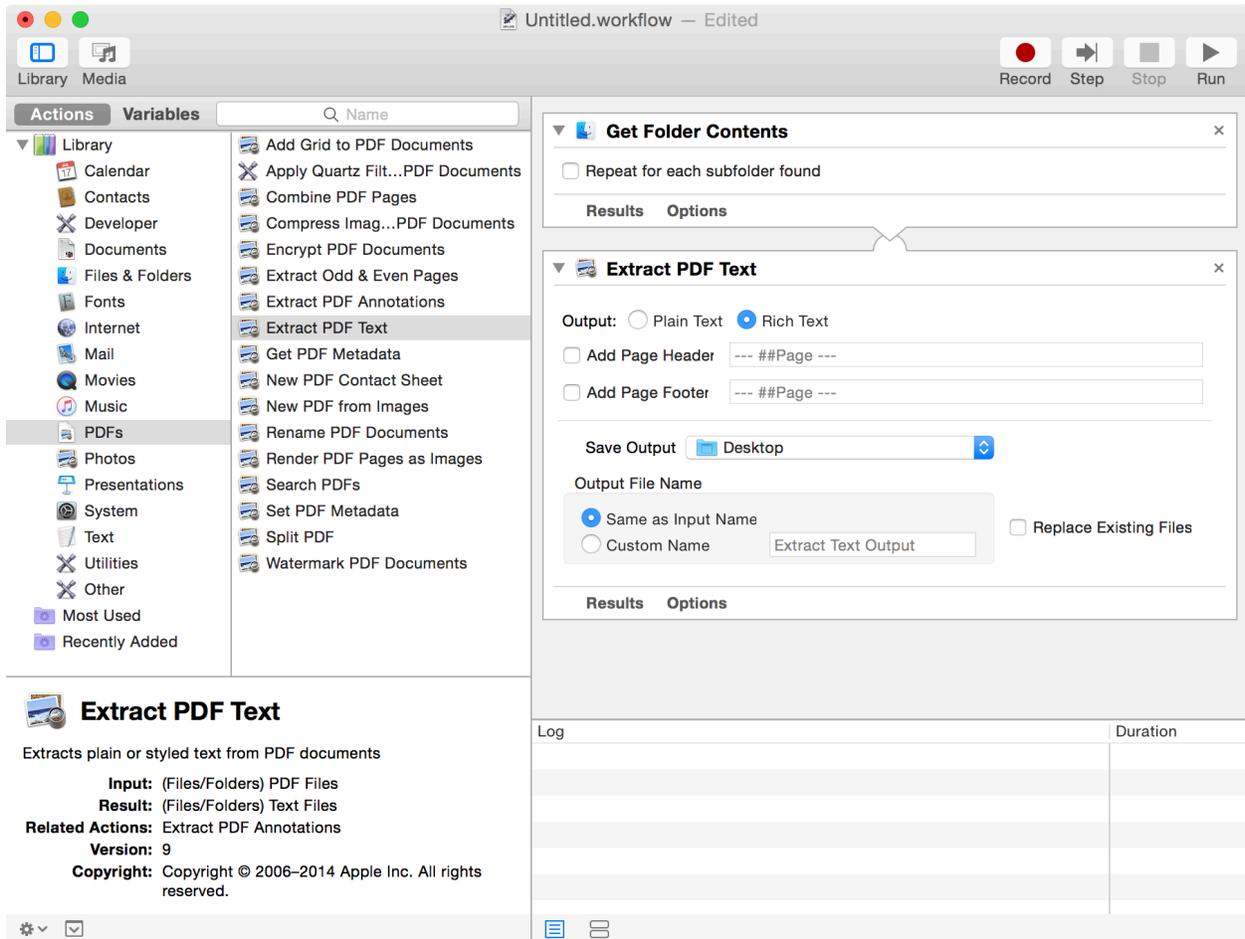


Figure. Screen shot of the Mac OS X Automator app Workflow for converting PDFs into RTF.

Metadata files: Overview

EMERSE was designed to incorporate data from electronic health records. As such, it was designed to include documents that are connected to patients via a medical record number (MRN). Documents can have various sources (e.g., Radiology, Pathology, Epic, etc), and each document might have different metadata depending on the source (e.g., date, author, document type, clinical service, authoring clinician, etc). EMERSE also expects to have a list of all patients for which documents are loaded.

While this was the intended use, it is still possible to include documents even if they are not actually tied to a known patient. In such a case, when setting up the data import you will need to essentially create a fake “patient” MRN to connect to the document as well as filler data about the patient name, date of birth, gender, etc. This should not affect the performance of the system.

Following are details about the data files needed to import your own data into the VM, their expected structure, and contents. Four files in Microsoft Excel (.xlsx) are required to define the metadata, and a separate folder (simply called `documents`) is required to hold the actual documents that will be indexed. There are four components for which you need to provide data: the document source names, the metadata about the document sources, the document metadata, and the patient data. These four files are called:

```
documentMetadata.xlsx
patientMetadata.xlsx
sourceMetadata.xlsx
documentSources.xlsx
```

A fifth file is required by the loading scripts, but does not require any editing. It is called `solrmap.xlsx` and it should already be located on the virtual machine in `/app/data/emerse_pdi_job/`. In the event that it is missing or deleted, it can simply be re-created by pasting the table below into an Excel spreadsheet and naming it `solrmap.xlsx`

| SOLR Field | Excel field |
|----------------|-------------------------|
| ID | documentId |
| LAST_UPDATED | documentLastUpdatedDate |
| ENCOUNTER_DATE | documentServiceDate |
| ADMIT_DATE | admitDate |
| DISCHARGE_DATE | dischargeDate |
| CASE_ACCN_NBR | accessionNumber |
| DOC_TYPE | documentType |
| SVC | service |
| CSN | encounterNumber |
| DEPT | department |
| CLINICIAN | providerName |
| STATUS | status |
| DESC | description |

Note: Because of the way that the import scripts were written, they will only work if the column headers in the Excel files are not modified in any way. That is, the header names should remain the same and the order of all of the columns should remain the same. Changing the column order will break the import scripts.

Also note that all dates should ideally be in the MM/DD/YYYY format to ensure there is no ambiguity when importing the dates, especially around any 2-digit year abbreviation.

Optional Directions for installing OpenOffice:

It is probably easiest to create these Excel files on your own computer and then move them to the virtual machine (details for moving the files are provided within this document). However, if there is a desire to create, open, or edit Microsoft Excel xlsx files on the Linux virtual machine, this can be done using a free program called OpenOffice, which can be found at: <http://www.openoffice.org>

Installing this application would involve downloading the file, which likely will be downloaded to the Downloads folder.

Download the full installation [Linux 64-bit (x86-64) (RPM)]. (Your VM will need to be able to access the Internet for the download to occur).

In the Terminal app, type the following commands in this order (password demouser may be required):

```
cd /home/emerse/Downloads/  
tar -xvsf Apache_OpenOffice_4.1.3_Linux_x86-64_install-rpm_en-US.tar.gz  
cd /home/emerse/Downloads/en-US/RPMS/  
sudo rpm -Uvih *rpm
```

Note that the filename above starting with Apache_OpenOffice_4.3.1_Linux... might change depending on when it is downloaded since the versions may change over time.

Go to the Linux VM menu bar and choose: Home → Downloads → en-US → RPMS → desktop-integration:

Double click on the file that has the name “redhat-menus” in it, which might be something like: openoffice4.1.3-redhat-menus-4.1.3-9783.noarch.rpm

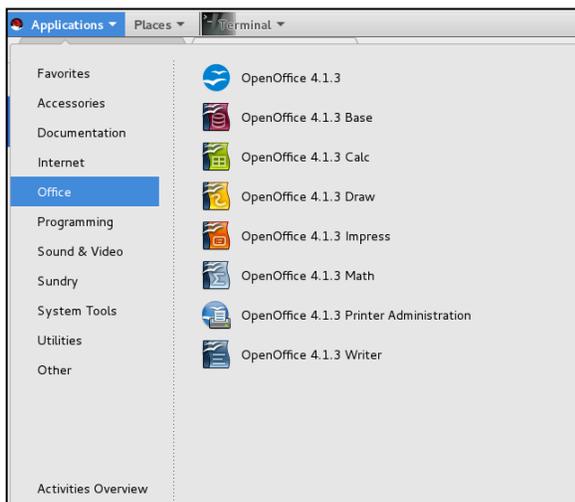
Click the blue button that says Install. When it is done, close that window. Do not click the red uninstall window:



□ It should then be possible to launch OpenOffice from the command line (Terminal) by typing:

```
openOffice4
```

Or, it the application also be launched form the menu bar, by going to Applications → Office → OpenOffice



Metadata files: Document sources

The first file to define is related to the document sources. In EMERSE, documents are displayed according to source system which helps distinguish where a document came from. In general, a source can be thought of as a source system, such as a radiology system, a pathology system, or the EHR itself. You can include just one source, or you can have multiple sources. These sources are defined in the file called:

`documentSources.xlsx`

There are 2 columns where the metadata about the document sources should be defined. Each is explained below. The order in which the sources are displayed in the EMERSE “Overview” section (from left to right) are defined by the order in which they are listed in this spreadsheet (from top to bottom).

| | |
|----------------|--|
| Source Key | This is used internally to link data between this <code>documentSources.xlsx</code> spreadsheet and the <code>sourceMetadata.xlsx</code> and <code>documentMetadata.xlsx</code> spreadsheets. The elements in this column can have any name, but there can be no spaces in the name provided, and each row must be unique. |
| Source Display | This is the name of the source as it will be displayed in EMERSE. This can have spaces in it, and would likely include things such as “Pathology”, “Radiology”, etc. (but without the quotes). |

Example `documentSources.xlsx` spreadsheet:

| Source Key | Source Display |
|------------|----------------|
| source1 | Epic EHR |
| source2 | Pathology |
| source3 | Radiology |

Metadata files: Document Source Metadata

For every document source there are, at a minimum, two required metadata fields (other than the text of the document itself). These are the date in which the clinical encounter occurred (often called the service date or encounter date), and the date when the document was last updated. However, each document source may have additional metadata that could be displayed for the users and can include items such as the clinical department, the clinician's name who authored the document, or even the document/note type (e.g., progress report, discharge note, etc.). The types of metadata are defined in this Excel spreadsheet which is called:

`sourceMetadata.xlsx`

This file contains 3 columns, described below. The order in which these elements are listed (top to bottom) are the order in which they will appear in EMERSE (from left to right), with the caveat that the text snippet showing any text found for a note will always appear on the left, and, the order is only considered within each document source.

| | |
|------------------------------------|---|
| <code>Source Key</code> | This is the same source that should be listed in the <code>documentSources.xlsx</code> file and the <code>documentMetadata.xlsx</code> file. The names cannot contain any spaces, and is only used internally to link between the files. |
| <code>nameFromDocumentSheet</code> | This provides a linking/map between the <code>sourceMetadata.xlsx</code> and the names of some of the column headers in the <code>documentMetadata.xlsx</code> file. It basically represents the various metadata elements that can be incorporated for each document source. (The names are fixed and cannot be changed, although this could potentially be changed by a developer who modified the underlying Solr schema.) |
| <code>displayName</code> | This is the name of the metadata element as it should be displayed in EMERSE. Often these will be names such as "Clinician", "Department", "Note Type", etc., but they can be anything. |

Note: There are two `nameFromDocumentSheet` elements that are required for each document source, although the display names are customizable. These two required fields are:

`documentServiceDate`
`documentLastUpdatedDate`

By Default EMERSE will sort data based on `documentServiceDate`.

The full list of elements for the `nameFromDocumentSheet` are defined in the table below. Note that these names are essentially placeholders, so even if a name appears to have a specific intended use, it can be overridden simply by providing a new display name. There are also two data types supported, `date` and `text`. Note that the table below is just for information to define what the elements are, it is not meant to be a spreadsheet used for importing data into EMERSE.

Table. Metadata elements defined in the `sourceMetadata.xlsx` sheet.

| nameFromDocumentSheet | Solr field name* | Required? | Type | Original intended use |
|------------------------------|-------------------------|------------------|-------------|--|
| documentLastUpdatedDate | LAST_UPDATED | Yes | Date | The date the document was last updated |
| documentServiceDate | ENCOUNTER_DATE | Yes | Date | The data of the clinical service that the document relates to |
| admitDate | ADMIT_DATE | No | Date | The admission date for the encounter to which the document belongs |
| dischargeDate | DISCHARGE_DATE | No | Date | The discharge date for the encounter to which the document belongs |
| accessionNumber | CASE_ACCN_NBR | No | Text | A field for a case accession number, used by the pathology department |
| documentType | DOC_TYPE | No | Text | The type of document, such as progress note or admission note |
| service | SVC | No | Text | The clinical service, such as rheumatology or cardiology |
| encounterNumber | CSN | No | Text | The encounter number to which the document belongs |
| department | DEPT | No | Text | The clinical department, such as pediatrics, or internal medicine |
| source | SRC_SYSTEM | No | Text | The name of the source system from which the document came. Note that this is different from the 'source key' defined elsewhere, so it could be re-purposed for another type of metadata element if desired. |
| providerName | CLINICIAN | No | Text | The name of the clinician who authored the document |
| status | STATUS | No | Text | The status of the document, such as Signed, Preliminary, etc. |
| description | DESC | No | Text | Any additional information that might be relevant about the documents |

* This is only provided for those wishing to look deeper into the underlying Solr configuration to understand how the fields are being mapped.

Example sourceMetadata.xlsx spreadsheet:

| Source Key | nameFromDocumentSheet | displayName |
|-------------------|------------------------------|-----------------------|
| source1 | department | Department Name |
| source1 | encounterNumber | Encounter ID |
| source1 | documentType | Doc Type |
| source1 | documentServiceDate | Document Date |
| source1 | documentLastUpdatedDate | Doc Last Update |
| | | |
| source2 | documentServiceDate | Document Date |
| source2 | documentLastUpdatedDate | Last Updated Date |
| | | |
| source3 | admitDate | Admission Date |
| source3 | dischargeDate | Discharge Date |
| source3 | documentType | Document Type |
| source3 | documentServiceDate | Date of Service |
| source3 | documentLastUpdatedDate | Document Last Updated |

Note: documentServiceDate and documentLastUpdatedDate are required fields for each data source. The others are optional.

Metadata files: Document Metadata

Every document indexed will have its own metadata which relates to the types defined in the `sourceMetadata.xlsx` sheet. For example, if `service` is defined in the `sourceMetadata.xlsx` sheet, then the specific metadata elements for a document could be 'cardiology', 'pulmonology', 'rheumatology', etc.

The file in which these document-specific metadata are defined is called:

`documentMetadata.xlsx`

Every row in this spreadsheet represents a single document to be indexed. Every document must have a unique identifier, and must be connected to a patient through a medical record number. The filename of the document to be loaded (from the `documents` folder) must also be specified here. For the metadata elements related to the documents that will be displayed in EMERSE, each element has its own column in the spreadsheet, regardless of whether or not any metadata for that element actually exists for the documents. The specific cells related to those columns should be left blank if no metadata exists. The following table details the elements present in this spreadsheet.

| Column Header Name | Required? | Description |
|--------------------------------------|-----------|--|
| <code>fileName</code> | Yes | The name of the document, located in the <code>documents</code> folder, including the file extension. |
| <code>source</code> | Yes | The source that the document belongs to, which would be one of the elements listed under <code>Source Key</code> in the <code>documentSources.xlsx</code> file |
| <code>patientMRN</code> | Yes | The patient medical record number, which must match a MRN in <code>patientMetadata.xlsx</code> |
| <code>documentID</code> | Yes | A unique identifier for each document. This can be anything, but must be unique for each document |
| <code>documentServiceDate</code> | Yes | See definitions in the table describing the metadata in the <code>sourceMetadata.xlsx</code> file |
| <code>documentLastUpdatedDate</code> | Yes | |
| <code>documentType</code> | No | |
| <code>department</code> | No | |
| <code>encounterNumber</code> | No | |
| <code>providerName</code> | No | |
| <code>admitDate</code> | No | |
| <code>dischargeDate</code> | No | |
| <code>accessionNumber</code> | No | |
| <code>service</code> | No | |
| <code>status</code> | No | |
| <code>description</code> | No | |

Example documentMetadata.xlsx spreadsheet:

| fileName | source | patientMRN | documentID | documentServiceDate | documentLastUpdatedDate | documentType | department | ...* | description |
|----------|---------|------------|------------|---------------------|-------------------------|----------------|--------------------|------|------------------------|
| abc.txt | source1 | 1000000056 | abc12345 | 12/26/16 | 1/8/17 | Nursing Note | Orthopedics | | |
| def.txt | source3 | 1000000045 | def43454 | | | | | | clinical trial subject |
| | | | | 12/26/16 | 1/8/17 | Discharge Note | | | |
| ghi.txt | source1 | 1000000034 | ddd33433 | | | Nutrition Note | Emergency Medicine | | |
| jkl.txt | source2 | 1000000056 | 12323abc | 9/21/16 | 10/6/16 | | Pediatrics | | |
| mno.txt | source2 | 1000000056 | 231325f3 | 10/7/16 | 10/17/16 | | Urology | | |
| pqr.html | source1 | 1000000045 | ddf4543 | 9/21/16 | 10/8/16 | Discharge Note | Neurosurgery | | |
| stu.html | source3 | 1000000045 | hhjuy654 | | | | | | clinical trial subject |
| | | | | 9/21/16 | 10/11/16 | Nursing Note | | | |
| vwx.html | source2 | 1000000023 | ddfdfdf12 | 12/12/16 | 12/13/16 | | Neurosurgery | | |

* Other columns should appear, as defined elsewhere in the previous table, but are not shown here due to space limitations. In addition, the order of the columns should not change, and all columns must be in this spreadsheet even if no metadata exists for them.

Note: The documents themselves (as defined under the `fileName` header column) should all be located within a directory called `documents`. This folder of documents should be in the same directory as the other metadata files when importing them into EMERSE.

Metadata files: Patient Metadata

Every document that is imported must be associated with a patient through a medical record number. A patient can have one or (more likely) many associated documents. Patients are defined in a simple Excel spreadsheet called:

`patientMetadata.xlsx`

In this spreadsheet every patient is listed in a row with some additional metadata about them including date of birth, race, gender, and, most importantly, medical record number (MRN). Currently race and gender are free text fields and there is no constraint on what can be entered for them.

Example `patientMetadata.xlsx` spreadsheet:

| LastName | FirstName | patientMRN | DOB | gender | race |
|-----------|-----------|------------|------------|--------|-------|
| CURRY | CODY | 100000001 | 02/23/2004 | male | Black |
| RICE | CHAD | 100000002 | 12/22/1985 | male | Other |
| DURAN | DEVIN | 100000003 | 05/05/2006 | male | Other |
| SANCHEZ | EARNEST | 100000004 | 03/11/1980 | male | Other |
| COLE | JONATHON | 100000005 | 12/16/1931 | male | White |
| RILEY | SALLY | 100000006 | 10/01/1977 | female | Other |
| HENDERSON | LEONARD | 100000007 | 05/16/1984 | male | Black |
| ALEXANDER | LELA | 100000008 | 07/08/1967 | female | Other |
| FIELDS | REX | 100000009 | 10/03/2000 | male | Black |
| HINES | ROSIE | 100000010 | 01/13/1999 | female | Black |
| SIMS | DARNELL | 100000011 | 07/31/1980 | male | Asian |

Setting up a shared directory for copying data to the VM

Note: The following directions were created for a Mac, but the process should be very similar for Windows. The directions below were based on those found at helpdeskgeek.com

<http://helpdeskgeek.com/virtualization/virtualbox-share-folder-host-guest/>

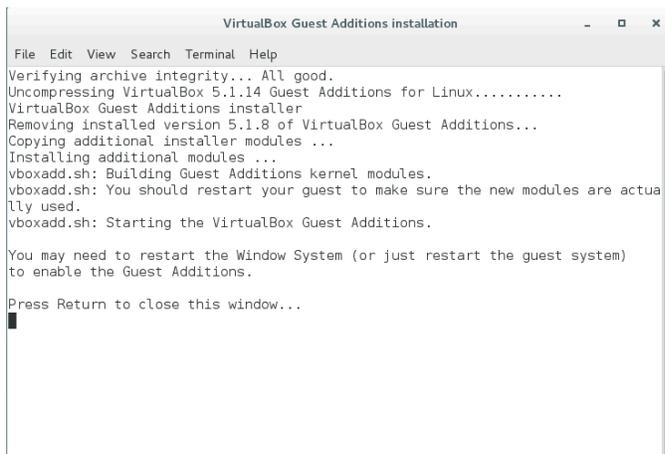
The virtual machine is essentially another computer (operating system) running on your computer. The VM is often called the *guest* and the main operating system on your computer is called the *host*. As such the *guest* OS has it's own disk space and can't "see" the files on your main (*host*) computer unless you set it up so that they can communicate. Once this is set up you can then copy files from the host OS to the guest VM. In these directions, the term "folder" and "directory" are used essentially interchangeably.

There are several ways to moves files to the VM for data import into EMERSE, and the directions below describe only one such approach. Depending on your setup, the default VM environment may not able to connect to the outside Internet, so using services connected to the internet for moving files may not be possible. There are ways to configure the VM to allow for internet connections, but this is not described here. Further, if the data destined for import contains sensitive patient information, it might not be ideal to move them via the internet anyway.

In the Virtual Box Application (not the Linux VM), go to the menu bar at the top of the screen and choose `Devices` → `Insert Guest Additions CD image...`

Then choose `Run` at the prompt. Enter the password `demouser`.

You will see a window with text about the installation. After it is complete it will show a message `Press Return to close this window...`, so press `Return`.



```
VirtualBox Guest Additions installation
File Edit View Search Terminal Help
Verifying archive integrity... All good.
Uncompressing VirtualBox 5.1.14 Guest Additions for Linux.....
VirtualBox Guest Additions installer
Removing installed version 5.1.8 of VirtualBox Guest Additions...
Copying additional installer modules ...
Installing additional modules ...
vboxadd.sh: Building Guest Additions kernel modules.
vboxadd.sh: You should restart your guest to make sure the new modules are actually used.
vboxadd.sh: Starting the VirtualBox Guest Additions.

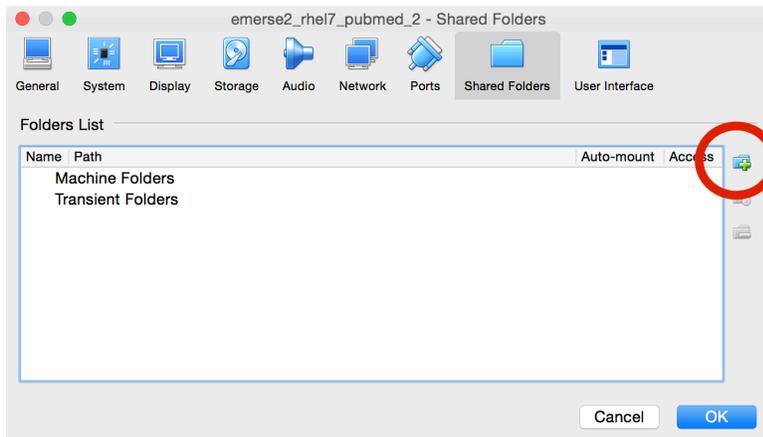
You may need to restart the Window System (or just restart the guest system)
to enable the Guest Additions.

Press Return to close this window...
█
```

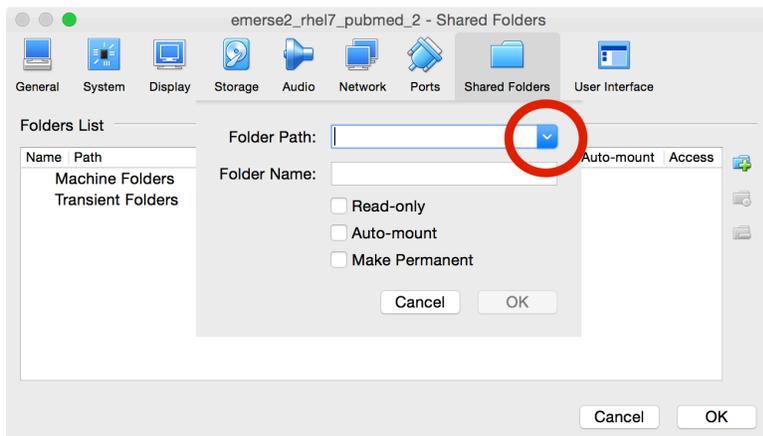
On the Mac Desktop create a folder where you will put all of the files that should be moved over to the VM. You can give it any name, but in this case call it: `vboxsharedfolder`

Go back to the VirtualBox Mac App and in the menu bar across the top of the screen choose `Devices` → `Shared Folders` → `Shared Folders Settings...`

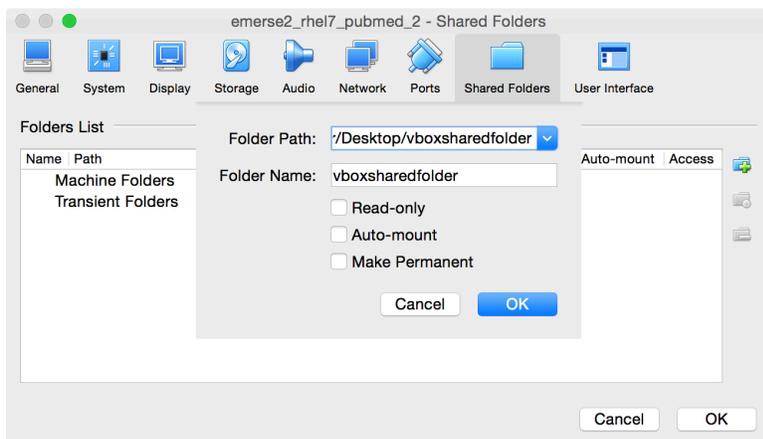
A window will open up, and then click on the folder icon with a plus sign



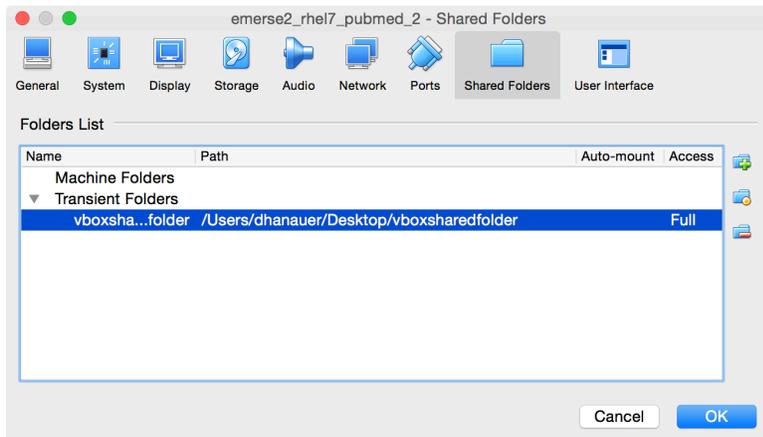
□ Under Folder Path click on the down arrow and select Other...



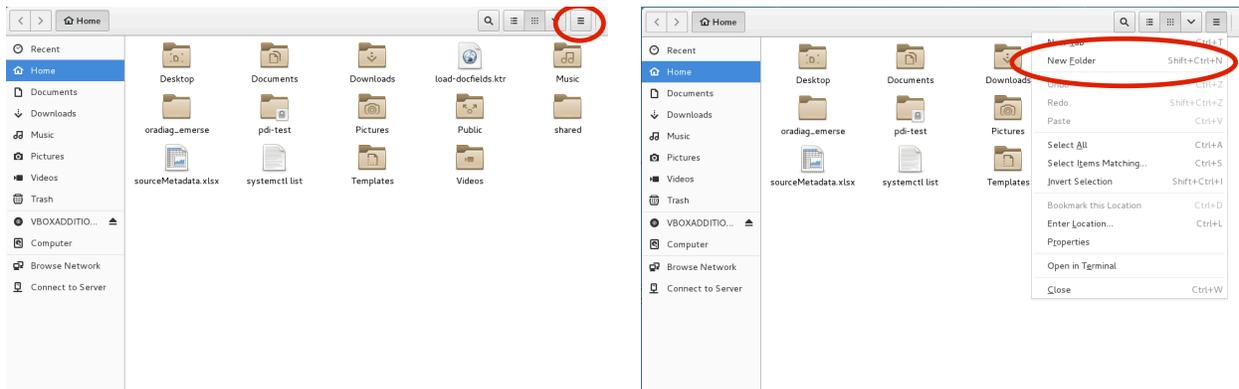
□ Navigate to the correct folder (the one named `vboxsharedfolder`) on the Mac Desktop, click on the folder once to select it, then click on the `Open` button. It should fill in the name of the folder where it says `Folder Name`. Then click `OK` to close that section and leave the other settings unchecked (e.g., `Read-only`, `Auto-Mount`, `Make Permanent`).



The folder will then show up in the table under the Transient Folders heading. Then click OK.

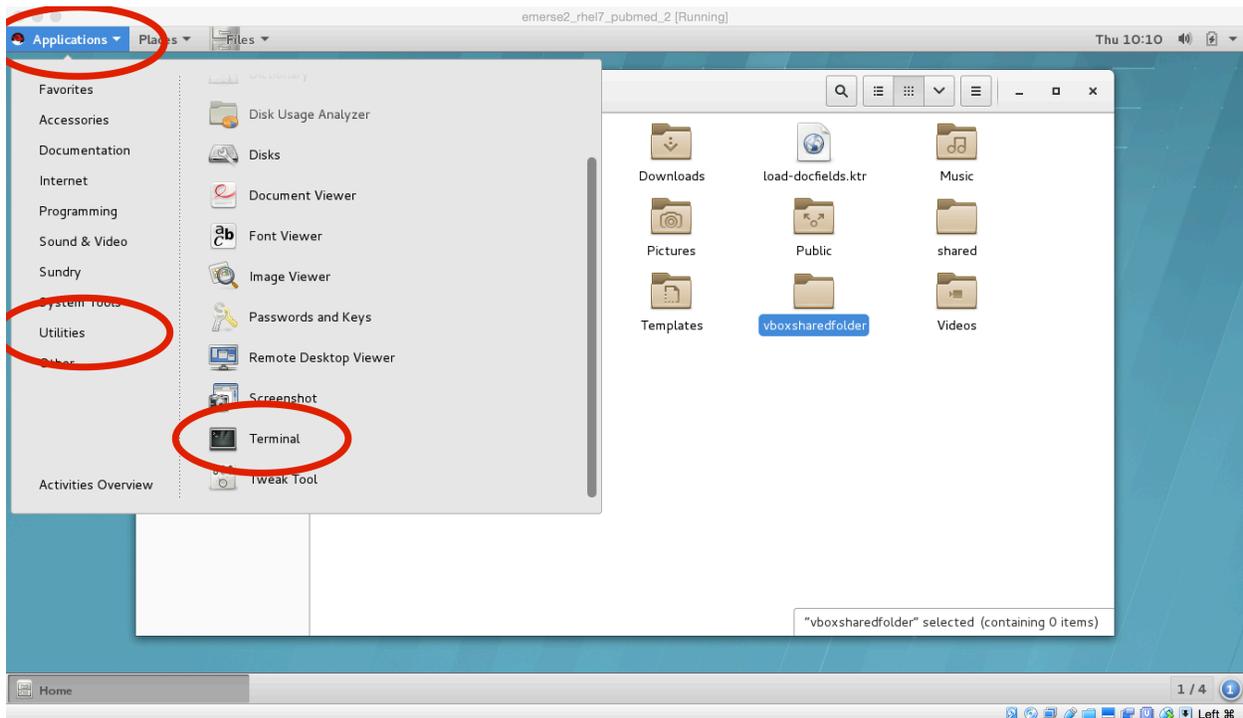


□ Switch to the guest OS (Linux) and navigate to the Home Directory by double-clicking on the Home folder on the Linux Desktop or choosing Places → Home from the menu bar. In the window that opens click on the icon with the horizontal lines in the upper right corner of the window and then choose New Folder.



□ Give it the same name as the one given for the folder on the Mac: vboxsharedfolder

□ In the Linux OS go to the menu bar and choose Applications → Utilities → Terminal



□ At the Terminal prompt type in:

```
sudo mount -t vboxsf vboxsharedfolder /home/emerse/vboxsharedfolder/
```

Then enter the password: demouser

Close the Terminal window.

At this point if you open the `vboxsharedfolder` on the Linux OS you should be able to see any files that are in the equivalent folder on the Mac Desktop.

Using the shared folder to copy the data files to the VM

Once the shared folder has been created, and the correct 'permissions' have been granted for access (see the prior section, "Setting up a shared directory for copying data to the VM"), you can then copy the data files to the VM. The directions for this are below.

□ Make sure the proper files have been moved to the `vboxsharedfolder` directory on the Mac, and then open the equivalent folder on the Linux virtual machine. The files that are on the host OS should now be visible in the folder on the guest OS (the Linux VM). Keep that folder open for now.

□ The next step is to then move those files to the right place on the Linux machine for importing. Go to the Linux menu bar and choose `Places` → `Computer`. Navigate (by double-clicking the folders to open then) to `app` → `data` → `emerse_pdi_job`

□ Drag the files from the Linux folder `vboxsharedfolder` to the `emerse_pdi_job` folder. These would include the files called:

```
documentMetadata.xlsx  
patientMetadata.xlsx  
sourceMetadata.xlsx  
documentSources.xlsx  
documents/ folder
```

If other files or folders with the same name are already there, replace them with the ones you are moving (or delete the older ones before copying over the new ones).

Make sure that the `solrmap.xlsx` file remains in the `emerse_pdi_job` folder. This file does not need to be edited but must be there for proper data import to occur. If it is not there, copy it there. Details about the file, and how to create a new one if necessary, are provided elsewhere in this document.

Using the scripts to load data into EMERSE

The directions in this section are based on the assumption that you have already created your metadata files for import and have the documents located in the `documents` folder as described in the previous sections of this manual. The following scripts will clear out the existing data in the system, load in the new data, and then index the documents so that they are ready to be searched. For this to work, you will need to first make sure that you have copied the updated metadata files into the `emerse_pdi_job` folder, and in that same `emerse_pdi_job` directory you have all of the documents that need to be indexed, located within a directory called `documents`. This includes:

```
documentMetadata.xlsx
patientMetadata.xlsx
sourceMetadata.xlsx
documentSources.xlsx
documents/ directory
```

□ Go to Applications → Utilities → Terminal

Navigate to the proper directory. To do this, at the command prompt type:

```
cd /app/software/PDI/jobs
```

□ Type the following five commands in the command line. Each will take a few second/minutes to run. They should be run in this specific order:

```
./runClearAll.sh
./runStagingTablesJob.sh
./runDocFields.sh
./runPatientLoadJob.sh
./runDocsToSolr.sh
```

□ Stop and then start Solr and Tomcat (password will be required for each command):

```
systemctl stop solr
systemctl stop tomcat
systemctl start solr
systemctl start tomcat
```

It may be necessary to wait 1-2 minutes until everything fully starts up again before trying EMERSE.

After the data are loaded, the patient count in EMERSE may not show up automatically. That count is updated using the Spring Scheduler within the app itself, and should auto-update about every 30 minutes. This also may require logging out of and then logging back into EMERSE for the change to be displayed in the application.

The virtual machine currently has the date range for the documents “hard-coded”, meaning that they will not change even if the actual imported documents have a different date range. This is a configuration option and can be changed. This won’t matter unless a search was done for a specific date range, otherwise EMERSE will always search across all documents regardless of

dates. Nevertheless, it is possible to change this so that the system will dynamically change the date range to accurately reflect the dates of the documents. The following directions detail how this can be done:

□ In the menu bar of the Linux VM, choose `Places` → `Computer`

□ Then navigate to the folder:

`app` → `software` → `emerge` → `apache-tomcat-8.0.33` → `webapps` → `emerge` → `WEB-INF` → `classes`

Note: the version of tomcat (8.0.33) may vary as we update the system to newer versions so that specific directory name may change from time to time.

□ Find the file called `project.properties` and double click it to open it in a text editor

Find the two lines:

```
batch.updateIndexMinDateFromSOLRIndex=false
batch.updateIndexMaxDateFromSOLRIndex=false
```

The first line represents the command to automatically update the minimum date range of the documents, and the second line represents the command to automatically update the maximum range of the documents. Set both of them to true, as in:

```
batch.updateIndexMinDateFromSOLRIndex=true
batch.updateIndexMaxDateFromSOLRIndex=true
```

Then, save the file.

□ Restart the Tomcat webserver by going to the Terminal application and typing:

```
systemctl stop tomcat
systemctl start tomcat
```

The password `demouser` will be required after issuing the stop and start commands.

The code that reads this properties file and updates the dates is embedded in the Java Virtual Machine (JVM) so it can't be run automatically. However, it should update on its own about every 30-60 minutes once the properties file has been edited and saved, and Tomcat has been restarted.

SECURITY CONSIDERATIONS

The virtual machine that EMERSE runs on or may or may not be able to access the Internet depending on how it, and the host Virtual Box application, is configured. If real patient data were to be loaded into the VM, it will be important to understand this configuration to make sure the security settings are in compliance with your institution's policies. It may be important to contact your local IT department to obtain guidance about how the VM is configured to ensure that the data are kept secure.

If the networking on the VM is set to NAT mode, then it is basically "sandboxed" and invisible to the outside world, with any exceptions defined in the NAT port forwarding table. However, if the VM is in bridged mode then the VM would be completely visible to the outside world since it essentially would have its own IP address on the network. If the VM (and host computer) is running behind an institutional firewall, additional protections may be in place because of where it is running, but that protection would not be available if the host machine were running outside of that firewall. Again, consult with your local IT group on advice about the best way to configure the VM to ensure security.

Of course, one way to reduce exposure to the outside world is to disconnect your host machine (the computer running the VM) from the Internet entirely (unplug it from Ethernet, turn off wireless, etc). When accessing EMERSE from the VM, a connection to the Internet is not necessary since everything runs locally.

Also, note that the VM ships with a standard username and password for all aspects of the system (Linux, EMERSE login, Oracle database, etc). These should be changed if real patient data were loaded into the system.

Note that some of the default passwords required by the application are located in the `project.properties` file which is itself located in:

```
/app/software/emerse/apache-tomcat-8.0.33/webapps/emerse/WEB-INF/classes/
```

MAKING EMERSE AVAILABLE ON THE HOST MACHINE

There may be cases in which you want to let EMERSE run on the virtual machine (the *guest* operating system) but would prefer to access EMERSE via the browser on your primary machine (the *host* operating system). This is possible to do, but requires some configuration. Further, this may make the system more accessible to the outside world, so it is important to consider any security implications about changing how the system might be accessed from outside the VM itself.

For those wishing to proceed, here are the directions, assume Virtual Box is already installed:

Launch Virtual Box

Install the VM VirtualBox Extension Pack. This can be found here:

<https://www.virtualbox.org/wiki/Downloads>

Download the file, which has a `.vbox` extension, and then double-click on it to install it.

In the VirtualBox app itself (not a virtual machine), go to `VirtualBox` → `Preferences`

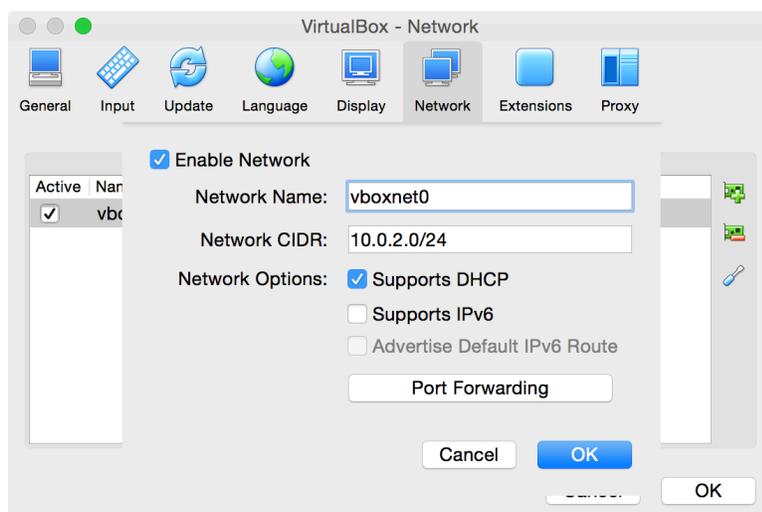
Go to the `Network` Tab

Right click in the table, or click on the Plus icon to `Add new NAT network`

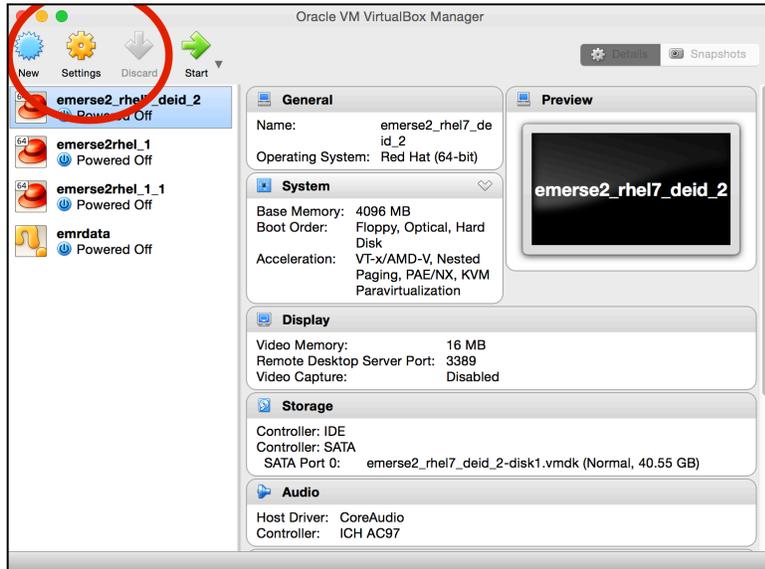
Can give it any name, but here we use `vboxnet0`

Can leave the other settings the same for now (will ignore `Port Forwarding` for now)

Click `OK`



□ In the main VirtualBox application, click once on the correct virtual machine in the list to select it, and then click on Settings.



□ In the Settings section,

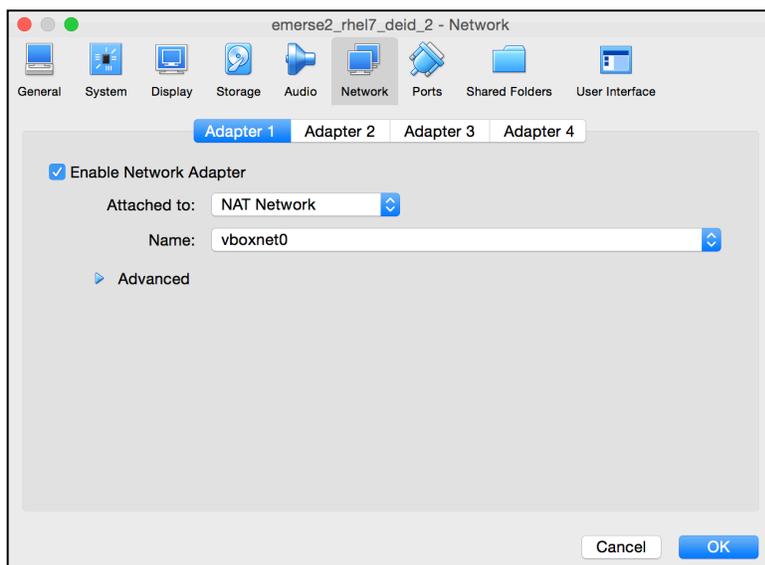
Make sure “Enable Network Adapter” is checked

Set it to, “Attached to: NAT Network”

Under Name, select the name given to the Network Name above, in this case vboxnet0

Can leave Advanced unchanged.

Click OK



□ Start the VM by selecting it in the VirtualBox app, and the clicking the green Start arrow.

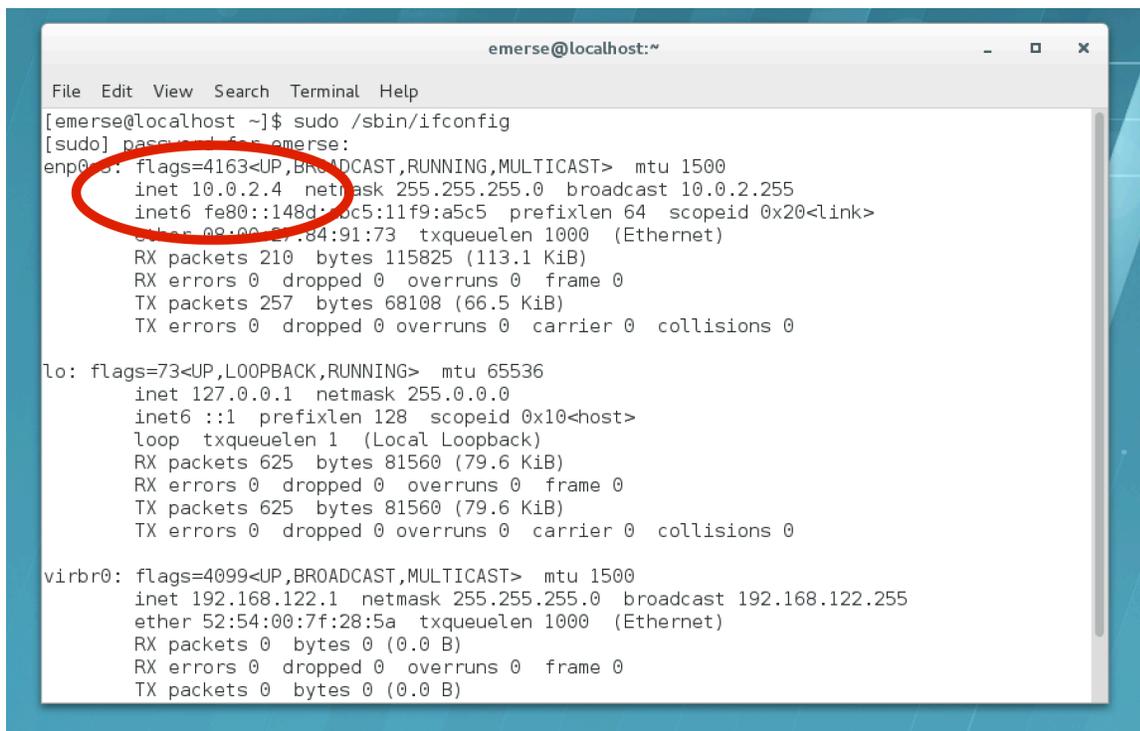
□ Login to the VM (username: emerse, password: demouser)

□ Go to Utilities → Terminal

Type: `sudo /sbin/ifconfig`

□ Write down the IP address shown near the `enps03`:

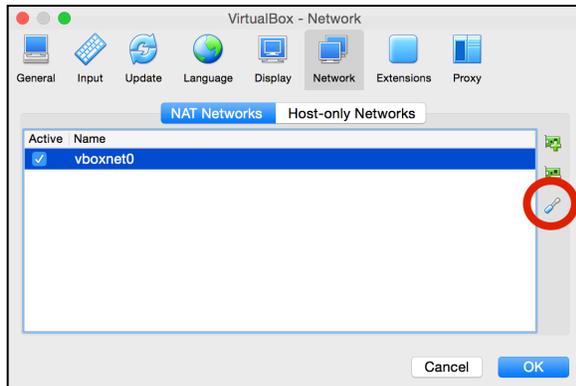
In the example screen shot below it says `inet 10.0.2.4`



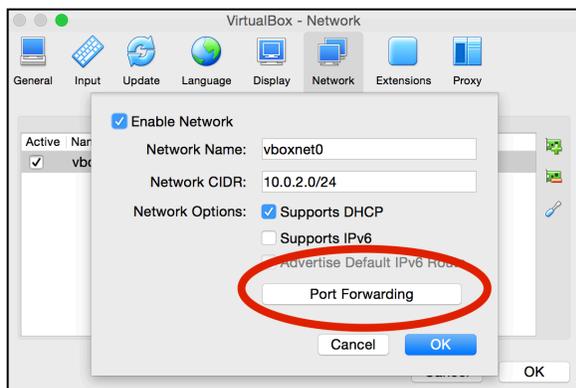
```
emerse@localhost:~  
File Edit View Search Terminal Help  
[emerse@localhost ~]$ sudo /sbin/ifconfig  
[sudo] password for emerse:  
enps03: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
inet 10.0.2.4 netmask 255.255.255.0 broadcast 10.0.2.255  
inet6 fe80::148d:bc5:11f9:a5c5 prefixlen 64 scopeid 0x20<link>  
ether 08:00:07:84:91:73 txqueuelen 1000 (Ethernet)  
RX packets 210 bytes 115825 (113.1 KiB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 257 bytes 68108 (66.5 KiB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
inet 127.0.0.1 netmask 255.0.0.0  
inet6 ::1 prefixlen 128 scopeid 0x10<host>  
loop txqueuelen 1 (Local Loopback)  
RX packets 625 bytes 81560 (79.6 KiB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 625 bytes 81560 (79.6 KiB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
virbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500  
inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255  
ether 52:54:00:7f:28:5a txqueuelen 1000 (Ethernet)  
RX packets 0 bytes 0 (0.0 B)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 0 bytes 0 (0.0 B)
```

□ Go back to the VirtualBox VM application (not the VM itself) and choose `Preferences...` and got to the `Network` tab, and in that table go to the `NAT Networks` section

□ Go to the `Edit` section for the NAT Network `vboxnet0` by right-clicking to `Edit NAT Network`, or by clicking on the small screwdriver icon:



□ Click on the `Port Forwarding` Button:



□ Make sure, the `IPv4` pane is selected, then,

For `Name`, can enter anything, such as `EMERSE Tomcat`

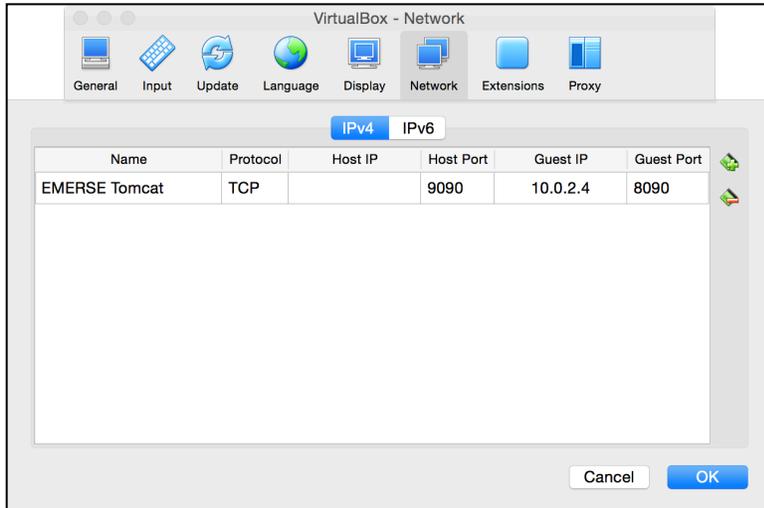
Set `Protocol` to `TCP`

For `Host IP`, leave it blank

For `Host Port` enter `9090`

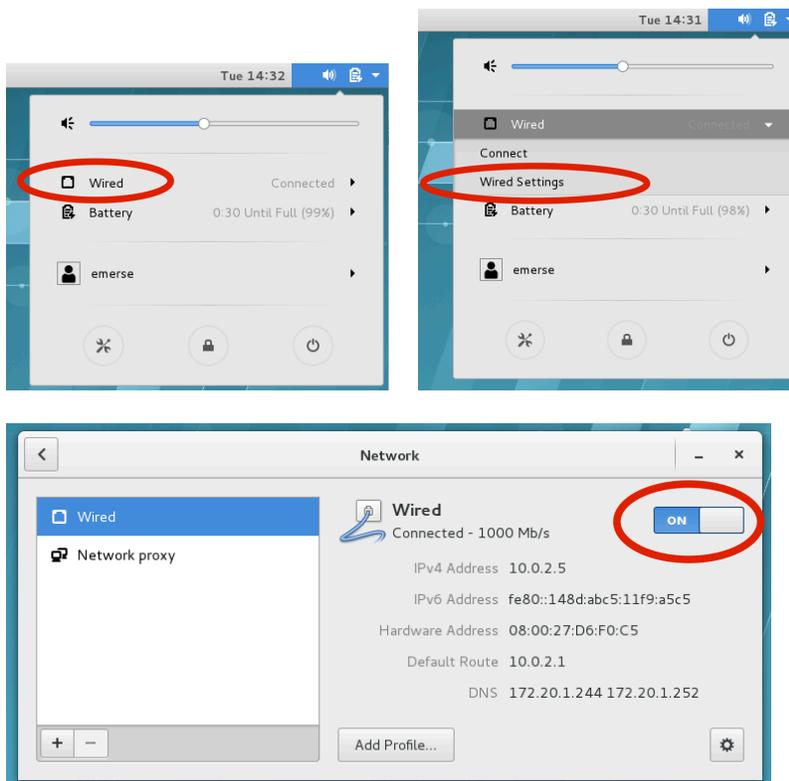
For `Guest IP` enter in the prior IP address that was written down (in this example it is `10.0.2.4`, but that may not always be the case, and it may even change from time to time but hopefully not).

For `Guest Port` enter `8090`



□ If the actual EMERSE VM is running, it may be necessary to turn the network settings off and then on again to have them change:

In the upper right of the VM screen click on the icons, then Wired → Wired Settings, then turn the connection OFF, then ON again.



At this point should it should now be possible to go to the browser on the host machine (not the VM) and enter the URL:

`localhost:9090/emerse/`