

cityEHR – Installation Guide

Open Health Informatics

John Chelsom

27/09/17

Table of Contents

1.Introduction.....	3
2.Basic Installation of cityEHR.....	4
2.1.Apache Tomcat	4
2.2.cityEHR Application.....	6
2.2.1.Standalone installation.....	6
2.2.2.Installation on a Server	6
2.2.3.cityEHR Source Code.....	7
3.Standalone Installation of eXist Database.....	8
3.1.Installation.....	8
3.1.1.Base eXist Installation.....	8
3.1.2.Running eXist as a Service.....	9
3.1.3.Configuring cityEHR.....	10
3.2.Database Backup.....	10
3.3.Database Recovery.....	11
3.4.Installing Pre-packaged Databases.....	11
3.5.Running eXist Under Tomcat.....	12
4.Integration Engine.....	13
4.1.Mirth Connect.....	13
5.Other Applications Used in Workshop Exercises.....	16
5.1.Open Office.....	16
5.2.Firefox Web Browser.....	16
5.3.Protégé 4.1.....	16
5.4.Notepad ++.....	17
5.5.Komodo Editor.....	18
6.Configuring the Applications.....	19
6.1. Mirth Integration Engine.....	19
7.Starting and Stopping Tomcat.....	21

Revision History

Date	Version	Changes	Owner
15/01/15	1.0	First Version, derived from Workshop Installation Guide	JC
15/05/15	1.1	Includes details of standalone database installation	JC
25/10/15	1.2	Includes details of running database under Tomcat	JC
26/06/17	1.3	Updated for cityEHR 1.5, including new configuration for standalone eXist database	JC
27/09/17	1.4	Updated for cityEHR 1.6	JC

1. Introduction

This document describes the installation of the open source applications required to run the cityEHR health records system, including tools used for creating and maintaining information models.

cityEHR is a server-side web application which can be run through the Mozilla Firefox web browser on any machine that has network access to the server.

Note that cityEHR should be run using the Firefox web browser on client machines.

cityEHR can be run using its built-in database which is suitable for demonstration and evaluation purposes. For operational use with real live patient data it is recommended to use the stand-alone database configuration which is more robust and is more suited to performing daily backup.

The steps for a quick installation using the built-in database are:

1. Check/install the Java Runtime Environment (JRE 7 or 8)
2. Install Tomcat application server as a service on (for example) port 8080
3. Deploy the cityEHR application as cityehr.war

To install and configure a stand-alone database for running with live patient data, the additional steps are:

4. Install the eXist XML database
5. Configure eXist to run as a service on (for example) port 8022
6. Drop in the pre-packaged cityEHR database, or import the default information model
7. Configure cityEHR to use the stand-alone database

The most usual installation is on virtual servers, running Windows. For this the following specification of virtual server is recommended (two servers, one for Tomcat, one for eXist):

Quad core processor, 8Gb RAM, 200Gb disk, Windows Server 2008 R2 (or later)

To run cityEHR as an integrated system. Using HL7 v2 messaging you also need to install:

- Mirth Connect integration/messaging engine (only required for integration services)

To create information models that can be loaded to cityEHR you will also need:

- Open Office or Libre Office

The following applications are required only to complete the exercises in the cityEHR hands-on workshops - they are not required to run the cityEHR application

- Protégé ontology environment
- Komodo Edit (programmer's file editor - any other equivalent editor can be used)
- Notepad++ (alternative to Komodo, on Windows only)

These applications are available for installation on any platform (Windows, Mac, Linux) as described in this installation document below.

In addition, some other materials may be required for participants of cityEHR workshops; these are usually provided on a memory stick at the workshop and are also available for download at www.openhealthinformatics.org/resources - please check with your workshop co-ordinator for the latest version.

2. Basic Installation of cityEHR

2.1. Apache Tomcat

cityEHR is an enterprise Java application which must be deployed in a Java Application Server. Such as Apache Tomcat. If you already have Tomcat installed then you can add the cityEHR application as described in the next section – you don't need a fresh installation. If you have a different Java Application Server running (e.g. JBoss, Jetty, etc) then cityEHR should run successfully using that. The instructions below are for Tomcat 8, but you can use earlier versions (5.5, 6 or 7) if you already have them installed on your machine. The installation can be made on your local machine (which is assumed as the default below) or on a server.

You need to have the Java Runtime Environment set up on your machine before installing Tomcat. Generally this will be set up by default on Apple Mac and Linux distributions, but its not always there on Windows. To test whether the JRE is installed, open a command window and type: `Java -version` If installed you should get a response similar to:

```
Java version "1.5.0_20"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_20-b02-315)
Java HotSpot(TM) Client VM (build 1.5.0_20-141, mixed mode, sharing)
```

If you need to install the JRE then you can download a version for your operating system at <http://java.com/en/download/manual.jsp>

On Windows, follow the instructions at http://java.com/en/download/help/windows_manual_download.xml

On a Mac, follow the instructions at http://www.java.com/en/download/help/mac_install.xml

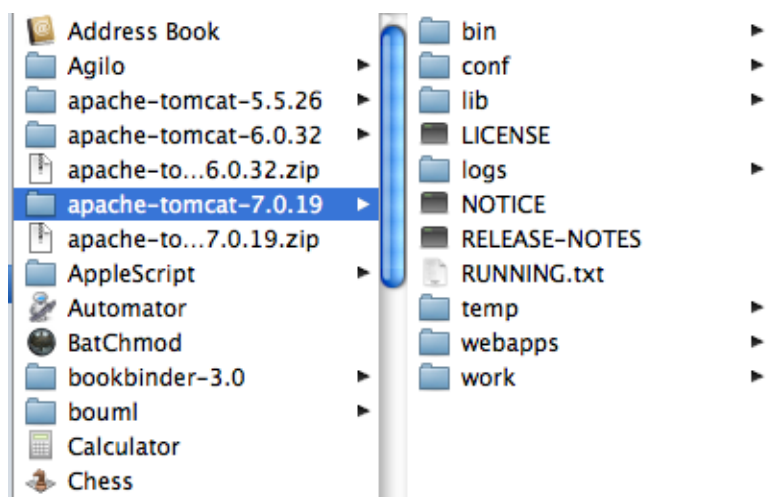
Tomcat can be downloaded from <http://tomcat.apache.org>

If you are running Windows you should use the 32-bit/64-bit Windows Service Installer which is a wizard that will guide you through the installation of Tomcat as a Windows service.

To avoid grief on Windows its best to install outside the c:\Program Files path, unless you are sure you have your Admin permissions set up correctly.

On other platforms you should select the core .zip file, download it and unzip the contents to a suitable directory you have created for it (e.g. called tomcat-7). We will refer to this directory as CATALINA_HOME.

Assuming you are installing from the .zip, then after unzipping you should now see a directory structure similar to the one below in CATALINA_HOME (screen shot is for Tomcat 7 on a Mac).

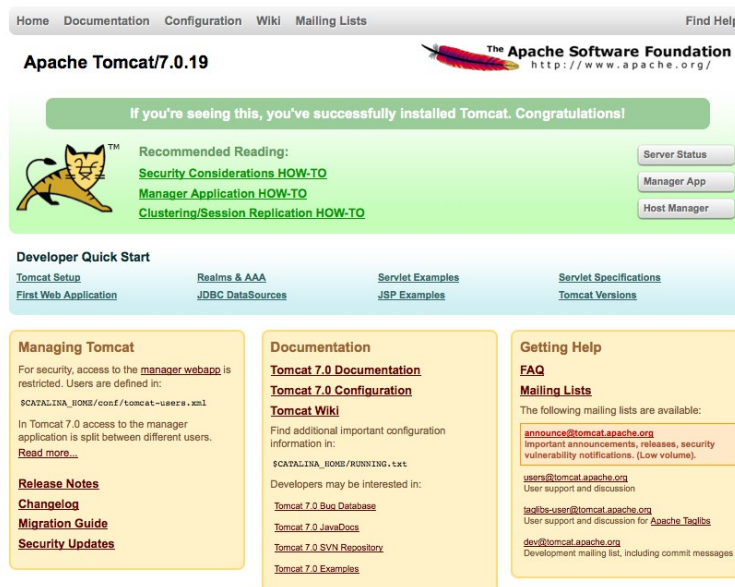


On Mac or Linux systems you may need to change the permissions on the files so that all users have read, write and execute permission - run `chmod -R 777` from the `CATALINA_HOME` directory. By default, Tomcat runs on port 8080 on localhost, so you may want to change the default settings in Tomcat (`CATALINA_HOME/conf/server.xml`) if you want to run it alongside existing servers.

You should now be able to start up Tomcat using the startup script provided with the installation in `CATALINA_HOME/bin` (`startup.bat` on Windows systems, `startup.sh` on Mac/Linux/Unix systems). To do this you will need to open up a command window, change to the `bin` directory and type the command. For more details on starting and stopping Tomcat see Section 7.

To check that Tomcat is running, open your web browser go to `http://localhost:8080` and you should see a page similar to the one shown below. If you have trouble, see the additional instructions in `CATALINA_HOME/RUNNING.txt` for how to resolve some common issues.

If you have installed Tomcat on a server (not your local machine) then you can access it from a web browser on a remote machine through `http://ipaddress:8080` where `ipaddress` is the IP address of the server (e.g. `192.168.1.1`).



If you have problems running cityEHR with the message that there is not enough Java heap space then you can increase it by adding the following to the `JAVA_OPTS` line in `/bin/catalina.sh` (Mac/Linux) or `catalina.bat` (Windows).

```
-XX:MaxPermSize=128m -Xmx512m
```

Good values for `-Xmx` are 1024, 2048, etc depending on the amount of memory available.

Alternatively, if you have Tomcat installed as a service in Windows, the `CATALINA_HOME/bin` directory should have a small windows application `tomcat7w.exe` (or equivalent for your version of Tomcat) that was created when tomcat was installed as a windows service. When you start this application it has a tab 'Java' where you can set the JVM parameters.

For most efficient operation you should set both memory pool options to the same value (e.g. 1024/1024, 2048/2048, 4096/4096, etc - depending on the amount of memory available).

To prevent Tomcat sessions from timing out, edit the `web.xml` file in `CATALINA_HOME/conf` directory (the default is 30 – i.e. times out after 30 minutes).

```
<session-config>
  <session-timeout>-1</session-timeout>
</session-config>
```

2.2. cityEHR Application

2.2.1. Standalone installation

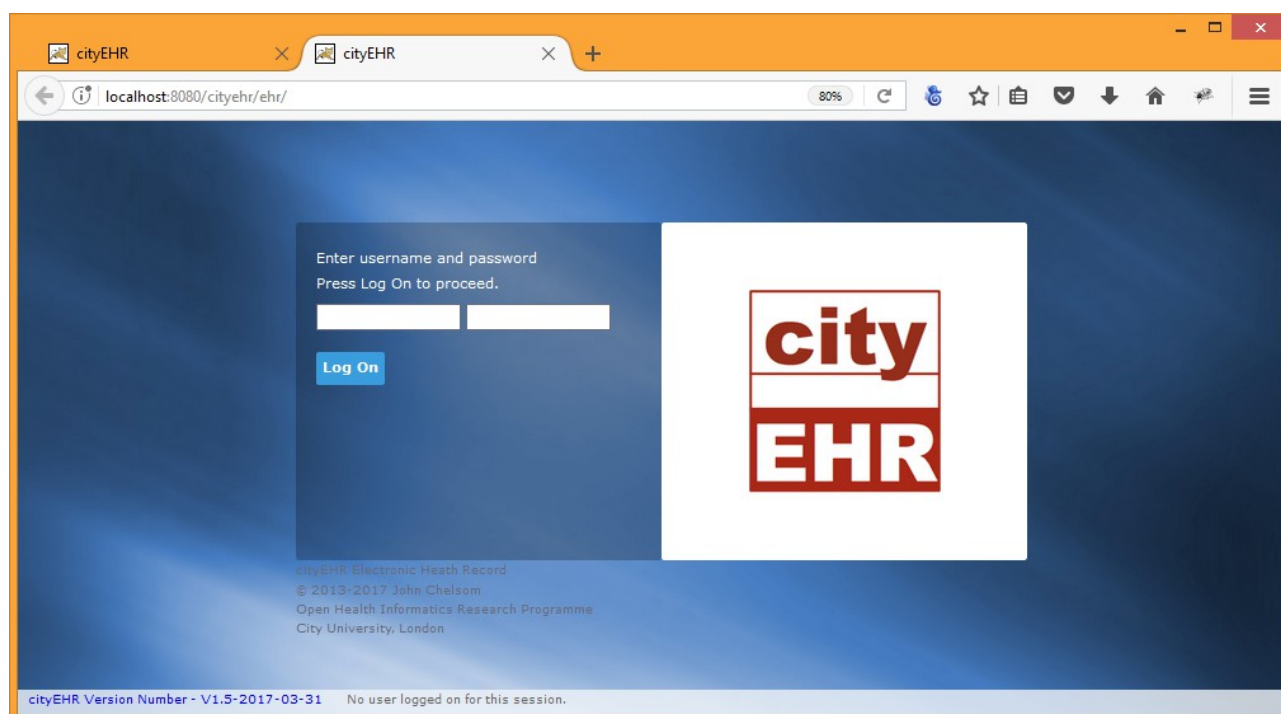
The latest version of cityEHR can be downloaded from <https://sourceforge.net/projects/cityehr/files>

The complete cityEHR is in the war file cityehr.war.

Stop the Tomcat service and copy the cityehr.war to CATALINA_HOME/webapps. Restart Tomcat and cityEHR will be deployed automatically (you may need to wait a minute this to happen)

The cityEHR distribution is configured by default to run on a standalone machine, using its built-in database which is pre-loaded with a demonstration EHR system.

So if you are running cityEHR on your local machine you should now be able to open your web browser and go to <http://localhost:8080/cityEHR> to see the log on screen of cityEHR similar to the one shown below. You can log on using the default user credentials, admin / password



2.2.2. Installation on a Server

The procedure nstallation on a server is exactly the same as the standalone installation.

After following the installation on a server with IP address 192.168.1.1 (for example), and assuming that Tomcat is running there on port 8080, you should be able to access the cityEHR homepage in a browser on any network machine using the URL [http:// 192.168.1.1:8080/cityEHR](http://192.168.1.1:8080/cityEHR)

If the homepage is not available then try to ping the server from the client machine to make sure the server is visible on the network. If the server is accessible but the homepage still does not display then check the firewall on the server is configured to allow traffic on port 8080.

Tomcat can be configured to run using SSL (encrypted) using the https protocol. For instructions on how to set this up, please refer to the separate guide - cityEHR on Tomcat With SSL.

2.2.3. cityEHR Source Code

cityEHR is written using various XML languages, which are plain text (i.e. non-compiled) code. The system can be described as an XRX application (XForms, REST, XQuery) although its code also includes a lot of XSLT and XHTML.

The XRX of cityEHR runs within an XForms application framework called Orbeon, which is an Enterprise Java application, running under Tomcat.

The cityehr.war which is deployed in Tomcat contains the Java code of Orbeon (which itself includes jar files for various XML processors, Apache FOP and the built-in eXist database). It also contains the XRX code of cityEHR itself.

Orbeon and eXist are open source, released under the LGPL license; cityEHR is also release under LGPL. This means that you can you use it in a similar way to a 'library' to create your own applications but you should not modify the source code, except to fix bugs.

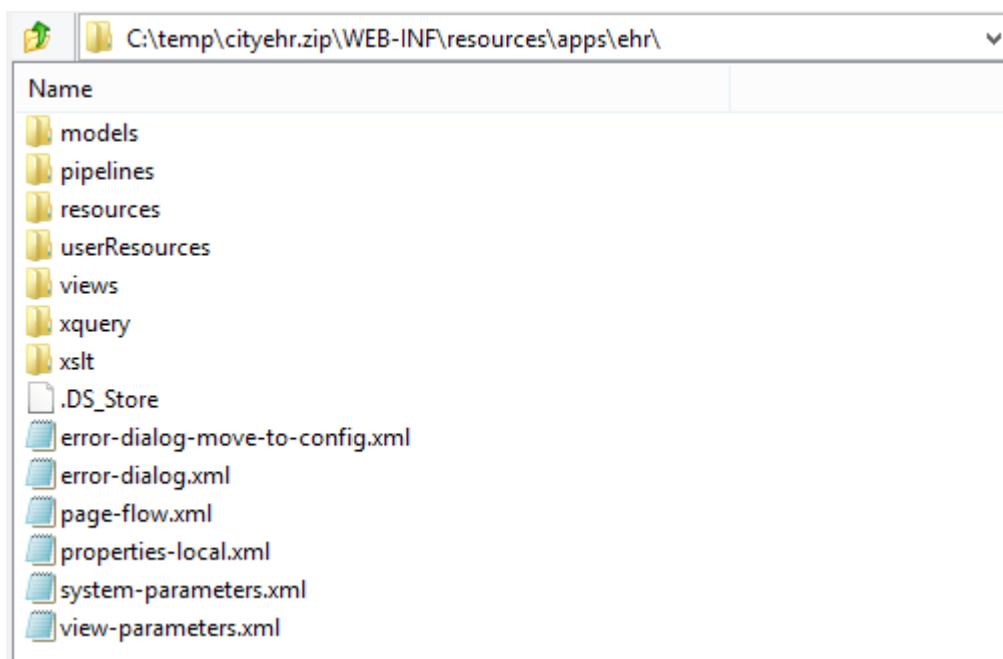
If you do modify the cityEHR code itself then you must make those modifications freely available as open source (LGPL) and include the original copyright notices in each file.

You can access the cityEHR source code once the cityehr.war file is deployed in Tomcat (from the CATALINA_HOME/webapps/cityehr directory) or rename cityehr.war as cityehr.zip and unzip it to access the code.

In either case, the cityEHR source code is located in:

WEB-INF\resources\apps\ehr

The contents of this directory should look similar to the figure below (on Windows)



3. Standalone Installation of eXist Database

The installation of cityEHR includes an instance of the eXist XML database (version 1.7) that is suitable for testing and experimenting with small scale installations (e.g. up to one thousand patient records). For larger scale, more robust deployment it is better to run a separate instance of the eXist database, which has the advantages of:

- Full text index for more efficient search
- Separate resource allocation (including working memory) and/or deployment on a different server to the Orbeon forms application
- Easy facilities for back up and recovery
- Built-in database management tools

3.1. Installation

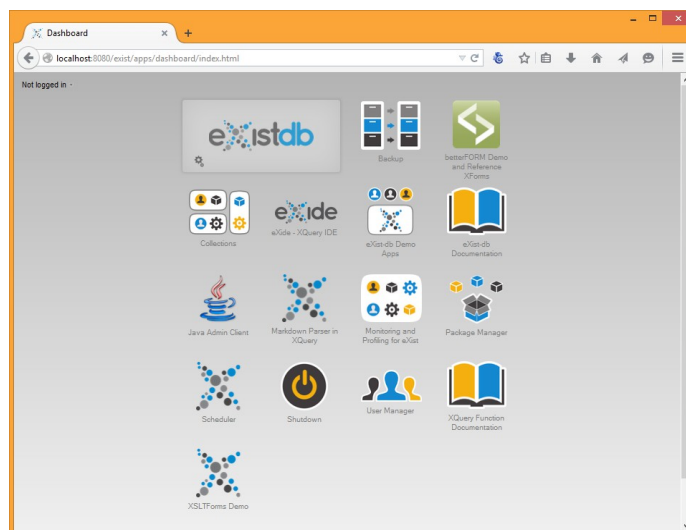
3.1.1. Base eXist Installation

It is recommended to use the stable eXist 2.2 release which is available for download at

<http://sourceforge.net/projects/exist>

The eXist database runs in its own servlet container (Jetty) and can be installed on the same server as Apache Tomcat, or on a different machine. Note that by default, eXist will use port 8080 and so will conflict with Apache Tomcat if you have installed that on its default port which is also 8080. The steps below will show you how to configure to run on a different port.

1. Make sure that Apache Tomcat (or any other service) running on port 8080 is shut down.
2. Run the eXist installer downloaded from sourceforge – this is a java installer so will run on any platform
3. During the installation you will be asked to specify the location for the database – make sure that on Windows systems this is outside the c:\Program Files directory. We will call this location <EXIST_HOME>
4. Once the installer has finished you can start the database by running <EXIST_HOME>/start.jar
5. This will start eXist on the default port 8080 and you can check the installation by opening <http://localhost:8080> in a web browser where you should see the eXist dashboard



6. Click the Shutdown button on the dashboard to shut down the database. You will be prompted to login – username is admin and the password is the one you specified during the installation process.
7. Check that eXist is no longer running by refreshing `http://localhost:8080` in the browser
8. You can also start and stop the database from the command line using the scripts in `<EXIST_HOME>/bin`
9. To change to port on which eXist is running, open the Jetty configuration file `jetty.xml` which is located in the directory `<EXIST_HOME>/tools/jetty/etc`
10. Change 8080 (e.g. to 8022) in the following XML element (its about line 38 in the file, or search for 8080)

```
<Set name="port"><SystemProperty name="jetty.port" default="8080"/></Set>
```
11. Now restart eXist by running `<EXIST_HOME>/start.jar` and check `http://localhost:8022` which should now bring you to the eXist dashboard
12. This completes the basic installation for eXist – if you are running the database on the same machine as Orbeon forms, you can now restart Apache Tomcat on port 8080 so that cityEHR is accessible.
13. By default, eXist will run with 1024Mb of Java heap space. If you launch eXist via one of the shell or batch scripts, you can increase this by changing the setting for `-Xmx` in the script.
14. On a Unix system, edit `EXIST_HOME/bin/functions.d/eXist-settings.sh`. Search for the `JAVA_OPTIONS` variable, which sets `-Xmx` for the server. The `CLIENT_JAVA_OPTIONS` variable does the same for the Java admin client. Instead of directly editing `eXist-settings.sh`, you may also override those variables globally in your own shell.
15. On Windows, the `-Xmx` settings is done in the main `.bat` files, .e.g. `EXIST_HOME\bin\startup.bat`.

3.1.2. Running eXist as a Service

1. To configure eXist to run as a service (rather than starting and stopping using the methods described above) you need to run an additional installation script
2. Make sure the database is shut down and then run the install script located in `<EXIST_HOME>/tools/wrapper/bin`
3. Note that configuration of the database service is in the XML file `wrapper.conf` located in `<EXIST_HOME>/tools/wrapper/conf`
4. To change the name of the service or the amount of memory allocated to the Java heap you will need to change the settings in `wrapper.conf`

```
# Initial Java Heap Size (in MB)
wrapper.java.initmemory=1024

# Maximum Java Heap Size (in MB)
wrapper.java.maxmemory=2048
```
5. Once the install script has run you can now start and stop eXist as a service (e.g. in Windows so to the Service Manager to do this)

3.1.3. Configuring cityEHR

With eXist running on port 8022 (for example) you can now configure cityEHR to use it. Log on to cityEHR as a system administrator (e.g. using the default account of admin / password) and select the Admin page from the top right of the system menu.

In the Admin page, scroll down the selections on the left hand pane and select Manage System Parameters.

In the section for Core Parameters, go to the entry for Database Configuration / Active databases. Change the settings for EHR and Audit log to be Standalone (installed). This is instructing cityEHR to use the database connection specified as Standalone (installed) in the Database Configuration / Installed databases entry which is directly below the Active databases entry (you may need to scroll down to see it).

By default, the Standalone (installed) connection is configured to run over port 8022 on the same server as the cityEHR application (so accessible as localhost). If necessary you can change the IP address and/or port number to the location of your eXist database.

The screenshot displays the 'Manage System Parameters' interface. The left sidebar contains a list of administrative tasks, with 'Manage System Parameters' selected. The main panel is divided into 'Dynamic Parameters' and 'Core Parameters'. The 'Core Parameters' section is further divided into 'Active databases' and 'Installed databases'. In the 'Active databases' section, the 'EHR' and 'Audit Log' are both set to 'Standalone (installed)', and 'Patient Access' is set to 'Local (built-in)'. The 'Installed databases' section shows a table with fields for 'server', 'port', 'username' (pre-filled with 'admin'), 'password', and 'credentials'.

Once the database is configured in Manage System Parameters, press the Save button (top right). You will be asked to confirm whether to save the parameters and restart the cityEHR system.

Once you have confirmed, the cityEHR home page will reload and you can log on again. Note that the new standalone database will be used for authentication, so only the default user (admin / password) will have access.

If the database is wrongly configured then after authentication on the home page you will be taken directly to the Manage System Parameters dialogue where you can make corrections and re-save the configuration.

3.2. Database Backup

It is recommended to make daily backups of the eXist database for live installations so that recovery can be made to the previous day's backup in the event of database failure. Database

back ups for eXist can be made using its built-in management tools, or simply by copying the database which is located at <EXIST_HOME>/webapp/WEBINF/data

The simplest way to backup with a simple copy is to create a script which runs each day during a time when the system is not in use. The steps to be taken are:

1. Shut down the Tomcat service
2. Shut down the eXist database service
3. Copy the directory <EXIST_HOME>/webapp/WEBINF/data to a back up location
4. Restart the eXist database service
5. Restart the Tomcat service

Windows command scripts which implement the above steps for daily and weekly back ups are available on the cityEHR distribution at sourceforge.net/cityehr.

Similar shell scripts for Linux and/or MacOS are easy to create following the example of the Windows command scripts.

3.3. Database Recovery

If the eXist database crashes or is terminated abnormally then it may not restart as expected. In such a situation, it is usually possible to recover the database by removing the log and lock files using the following procedure:

1. Ensure that the eXist database service is shut down
2. Make a copy of the *data* directory which is found in <EXIST_HOME>/webapp/WEBINF
3. In the *data* directory, delete the log file(s) *.log
4. In the *data* directory, delete the lock files *.lck
5. Restart the eXist service

To restore the database from a back up made using a simple copy:

1. Ensure that the eXist database service is shut down
2. Move the *data* directory which is found in <EXIST_HOME>/webapp/WEBINF to a different location
3. Copy the backup *data* directory to <EXIST_HOME>/webapp/WEBINF
4. Restart the eXist service

3.4. Installing Pre-packaged Databases

Some cityEHR applications may provide pre-packaged databases which are already configured with users, information models, cohort queries and sample data. These databases are specific to the version of eXist that is installed but can easily be configured to provide a quick start for a specific cityEHR application.

To install a pre-packaged database:

1. Download the pre-packaged database for the version of eXist that you have installed on your server (typically eXist 1.7 for the built-in database that ships with Orbeon, or 2.1/2.2 for standalone installations of eXist)
2. Shut down Apache Tomcat and the eXist database server (for standalone database installations only).

3. Move the eXist database to a backup location. The eXist database is located in:
CATALINA_HOME/webapps/orbeon/WEB-INF/exist-data shipped with Orbeon
<EXIST_HOME>/webapp/WEBINF/data standalone installations
4. If necessary, unzip the pre-packaged database to create directory called exist-data (for eXist 1.7 shipping with Orbeon) or data (for standalone eXist 2.1/2.2)
5. Copy the pre-packaged database to the same location on your server:
CATALINA_HOME/webapps/orbeon/WEB-INF/exist-data shipped with Orbeon
<EXIST_HOME>/webapp/WEBINF/data standalone installations
6. Restart the eXist database and Apache Tomcat

3.5. Running eXist Under Tomcat

It is possible to run the eXist database under Tomcat, using the same instance as the main cityEHR application.

You may want to use this configuration if you are running on a remote server that runs Tomcat, where you do not have access to install eXist directly in its own servlet container. This may be the case if you use a Tomcat hosting service from an ISP – this is more cost effective than running a hosted virtual server on which you have full control to run eXist separately.

To run eXist under Tomcat, you first need to make an exist.war file from the installed eXist distribution and then deploy that war under Tomcat.

4. Integration Engine

4.1. Mirth Connect

Mirth Connect is an open source health care integration engine. By supporting numerous standards (like HL7, EDI, XML, NCPDP, DICOM, and Delimited Text) and protocols (like LLP, JDBC, and FTP), Mirth Connect allows for the filtering, transformation, and routing of messages between disparate systems to allow them to share data.

A Mirth Connect installer is available as an executable file for Windows, Linux or Mac at <http://www.mirthcorp.com/community/downloads>.

Run the installer for your operating system and follow the instructions as they appear. Note that Mirth runs its own application server and database server. You must choose a port for the application server which is different from the port on which you are running Apache Tomcat e.g. if running Tomcat on the (default) port 8080, then you could select to run Mirth on port 2080 (which we will use for our examples).

The Windows installer also comes with the option to install and start a service which will run the background. If you choose to install the service, you also have the option of installing and running the Mirth Connect Server Manager which allows you to start and stop the service, as well as view the system logs and change the Mirth Connect backend database settings.

The default username and password for the login screen is admin and admin which for our purposes (unsecure) can be kept (you will be recommended to change the default during installation).

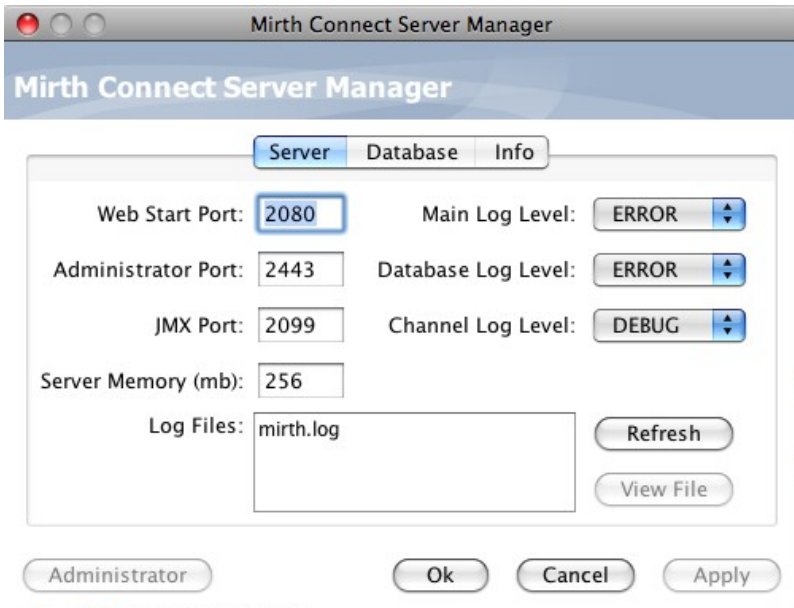
On Mac or Linux the Mirth server can be started and stopped by opening a command shell and typing the commands:

```
sudo launchctl start com.mirth.connect
sudo launchctl stop com.mirth.connect
```

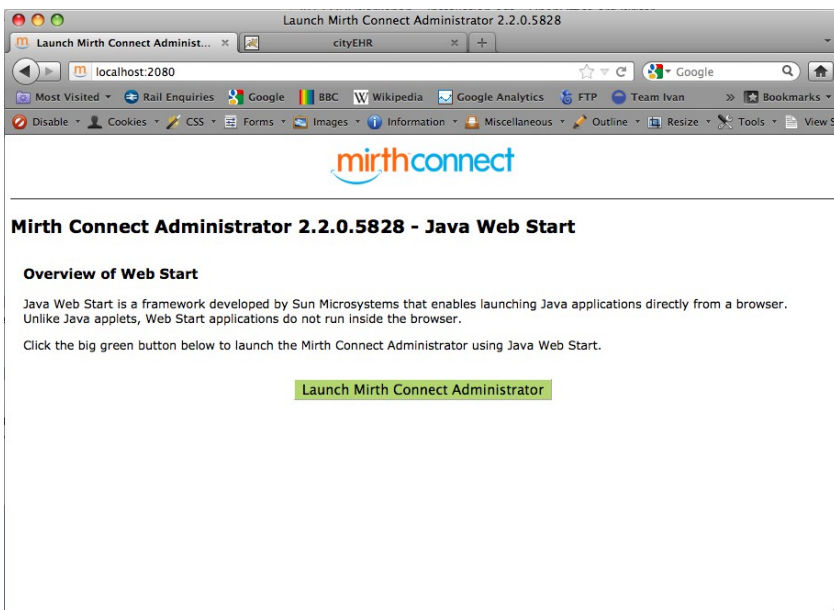
On a Mac you can automate the running of these commands as described in Section 7.

An optional Mirth Connect Shell can be installed which allows you to connect to a running Mirth Connect server through a command line interface. This tool is useful for recovery of a Mirth Connect installation.

After the installation, the Mirth Server Manager can be run by double clicking the mcmanager executable in the Mirth installation directory (this works on Windows, Mac or Linux) or (on Windows) you may have installed this on the startup menu. The Server Manager should look similar to this:

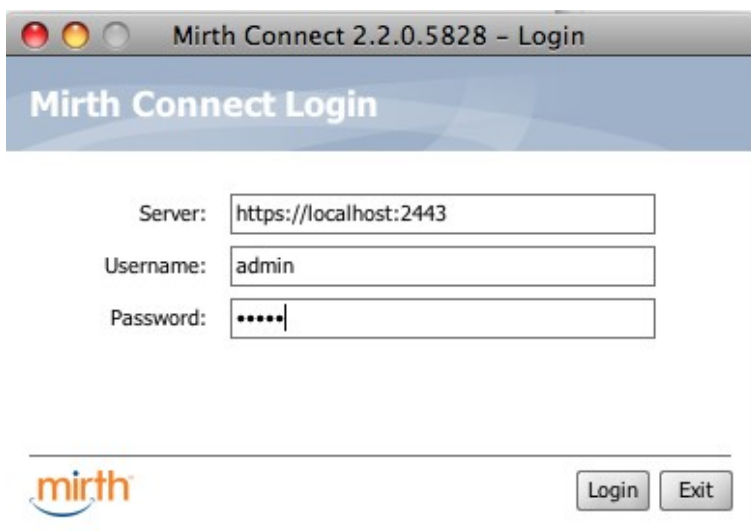


With the Mirth Server running (see above) you should be able to run the Administrator console by clicking the button in the Server Manager. Otherwise (on other platforms, or if the button is not available) you can launch the Administration Console directly from the server by accessing it in a web browser at: localhost:2080 (or whichever port you specified for your installation).

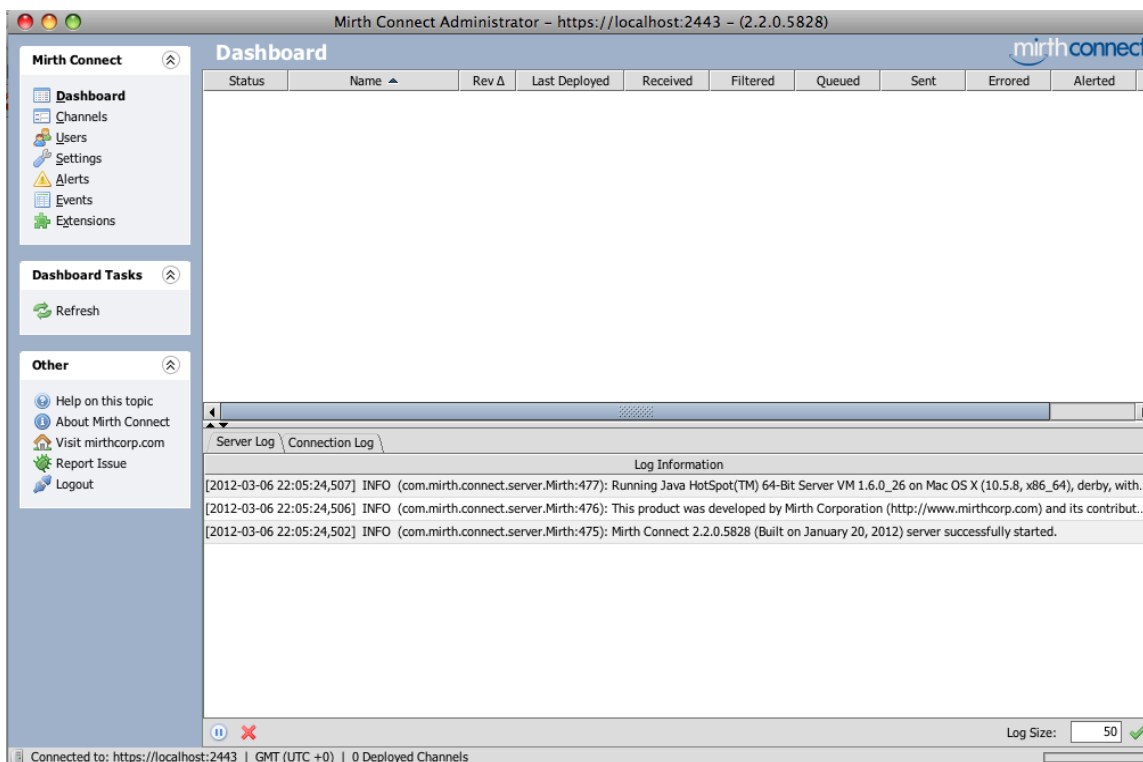


This will save the Java web start for Mirth which you can save in a location of your choice and then use in the future directly by double clicking on it.

You will get the log on screen similar to the one shown below, from which you can log in with username and password (defaults are admin / admin).



When running successfully, the Mirth Administration Console should look similar to the following:



After installing the basic Mirth environment there is some additional setup required for the cityEHR which is described in Section 6.1.

5. Other Applications Used in Workshop Exercises

5.1. Open Office

Open Office is a full office application suite with similar functionality to MS Office. Originally developed as the open source Star Office suite, Open Office was acquired by Sun Microsystems and subsequently Oracle. Following the acquisition of Sun by Oracle, and with the future of Open Office uncertain, the project forked to produce the independent Libre Office branch, with many of the original development team following this branch. Once Oracle had clarified its stance on Open Office some of these developers returned to the first branch, which Oracle transferred to the control of the Apache Software Foundation.

In the workshop we will use spreadsheets in the Open Office Calc application, which is equivalent to MS Excel. These spreadsheets are provided in .ods format which is one component of the XML-based Open Office Document Format (ODF) used by Open Office and some other office suites.

As part of the process of uploading our spreadsheets as configuration of the EHR system we will convert from ODF to MS Excel 2003 XML format.

As an alternative to Open Office you may try MS Excel and save in the 2003 XML format from there, although not all the features of our spreadsheets will work with Excel and the only testing that the workshop undergoes on this platform is by attendees at the workshop themselves.

To set up Open Office, go to <http://www.openoffice.org> and follow the instructions for download and installation on your platform. Note that the total application size for download is > 150Mb so you will need a decent Internet connection to complete successfully.

The current version used in the workshop is Open Office 3.3. You may use earlier versions if you already have them installed, but the workshop materials are no longer tested on those platforms, so we cannot guarantee they will work (but they probably will).

5.2. Firefox Web Browser

We are going to use the web browser to function as an XML processor, and for this its best to use Firefox with the Web Developer add-on installed. You can use any other modern web browser to process XML documents, but not all the steps in this workshop are guaranteed to work properly if you are using a different browser.

If you don't already have it, Firefox can be installed without affecting the operation of your existing browser. Web Developer can also be installed for the Chrome browser, so if you already have Chrome installed this is a good alternative to Firefox.

To install Firefox go to <http://www.mozilla.com/en-US/firefox/>

To install the Web Developer add-on, download and follow the instructions at:

Firefox: <https://addons.mozilla.org/en-US/firefox/addon/web-developer/>

Chrome: <https://chrome.google.com/extensions/detail/bfbameneiokkqbgdmiekhjnmfkcndhdm>

For convenience, we are going to use relative paths to locate files for stylesheets. This requires resetting a security option in Firefox, as follows.

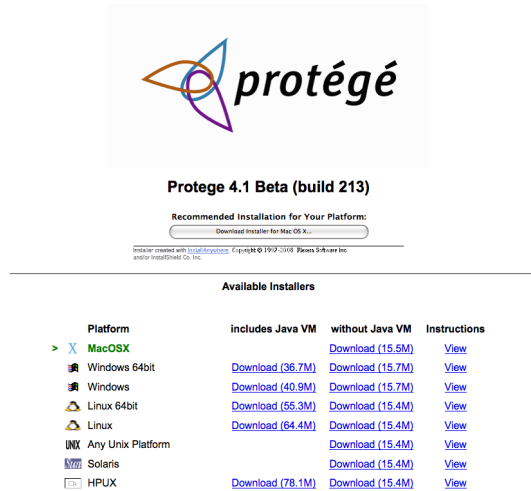
1. type `about:config` in the Firefox address bar
2. change the setting of `security.fileuri.strict_origin_policy` to false (by double clicking on it)

5.3. Protégé 4.1

Protégé can be downloaded from <http://protege.stanford.edu/download/registered.html>

Under the heading Protégé 4.1 release you should select the option for: Download Protégé - platform independent installer program. You should now be taken to a page similar to the one

shown below, from where you can download and run the installer for your platform (e.g. Windows, Mac, etc).



You should choose the download for your platform *without the Java VM*. Follow the on-screen instructions, or view the instructions on the web page by clicking the appropriate link. On successful installation you should be able to start up Protégé and see a first screen similar to the one shown below. If you have trouble getting Protégé to run on a Windows platform, then uninstall it (using the Windows Add/Remove programs utility) and then try any or all of the following:

1. Use the standard Windows version (not 64-bit)
2. Install in a location outside the c:\Program Files path
3. Use the installation that Includes the Java VM



5.4. Notepad ++

For Windows only, Notepad ++ can be downloaded from . Run the installation executable and follow the simple instructions. Once installed, download the XML Tools plugin by selecting Plugins | Plugin Manager from the main menu and checking the box for XML Tools, then press install.

5.5. Komodo Editor

The Komodo Edit tool is a programmer's file editor that can be used to edit XML files in their 'raw' state, showing all mark-up and content as plain text. The XML support built in to the editor helps with layout, identification of elements, attributes and content, tag completion, etc.

To download go to <http://www.activestate.com/komodo-edit/downloads> then select the installer for your platform and follow the instructions. Once successfully installed you should see a screen similar to the one below.

6. Configuring the Applications

6.1. Mirth Integration Engine

The basic Mirth integration service must be configured so that it can be used to import CDA documents to the cityEHR database. Follow the steps below to set up Mirth. It is assumed that Mirth is installed in the directory <Mirth_Home>.

1. Make sure that the Mirth service is stopped (stop it from the Mirth command console or using the command `sudo launchctl stop com.mirth.connect`)
2. The cityEHR stores its data in an XML database called eXist, which is installed as part of the cityEHR installation. In order for the Mirth integration service to import CDA documents directly into that database some additional code must be added to the basic Mirth installation.
3. Install Mirth's link to the cityEHR eXist database by placing the contents of the file `custom-lib.zip` (found in the `materials/Mirth` directory) into <Mirth Home>/`custom-lib`. That is, unzip the contents and move them to <Mirth Home>/`custom-lib`
4. Start Mirth service, and the cityEHR eXist database link will now be available behind the scenes (which means that the Mirth channel set up in the steps below will be able to import CDA documents directly into the cityEHR database).
5. Start the Mirth Administrator. On Windows this may have been installed on the startup menu during installation, or you can run the Java web start for Mirth which you should have saved in a location of your choice during installation (check the installation instructions if you can't find it).
6. Wait while the Mirth Administrator starts up (this can take some time, so be patient) and then press to Select Channels (on the left)
7. Select Edit Global Scripts then Import Scripts and select the `Scripts.xml` (found in the `materials/Mirth` directory). Update the eXist database settings in `Scripts.xml` here to match those of your local installation of cityEHR then save the changes. (If you have installed cityEHR running on port 8080 then there are no changes to make, if you have installed on a different port then edit the entry under the comment `// City EHR database RPC API URL`).

```
// City EHR database RPC API URL
globalMap.put ("cityehr.exist.database.rpc",
"xmldb:exist://admin:@localhost:8080/cityEHR/exist/xmlrpc");
```

8. The script is saved by pressing Save Scripts under the Scripts Tasks pane on the left.
9. There is a definition of a Mirth channel included in the materials. This is set up to watch a directory on your disk and import any files places there into the cityEHR database.
10. Before you enable this channel in Mirth you need to edit the location of the watched directories.
11. Open the file `cityEHR Import CDA.xml` (found in the `materials/Mirth` directory) in an XML or a text editor. Find the lines shown below and edit the directory paths to the ones you want to run on your installation

```
<property name="host">/Users/johnchelsom/Documents/workspace/mirth/Incoming/CDA</property>
<property name="moveToErrorDirectory">/Users/johnchelsom/Documents/workspace/mirth/Errors</property>
```

12. For example, on a Windows machine you might want to set the directory paths as follows:

```
<property name="host">C:\mirth\Incoming\CDA</property>
```

```
<property name="moveToErrorDirectory">C:\mirth\Errors</property>
```

13. Back in the Mirth Administrator select Channels (on the left)
14. Select Import Channel and select the following file from the materials/Mirth directory
cityEHR Import CDA.xml
15. Select the channel, then right mouse and Enable Channel
16. Select the channel, then right mouse and Deploy Channel
17. Select Dashboard (on left) this shows the deployed channels and how many messages have been received on each channel
18. Move (or copy) a sample CDA (e.g. CDA - SimpleEntry - BaseRegistration.xml from the CDA Samples directory in the materials/Mirth) into the watch directory that you set up in Step (11) and verify that it's been processed successfully (log information is displayed at the bottom of the Dashboard view or can be found in <Mirth Home>/logs.
19. You can check that the CDA has been imported into the database by starting cityEHR in your web browser, logging on and looking for the patient in Patient Search (e.g. the patient in the sample CDA registration above is Miss Nellie Stockton with a patient Id of 1234666666.
20. There is a second CDA document for Nellie Stockton called CDA - SimpleEntry - BloodPressure.xml which you can now import in the same way as above (move or copy the file to the watched directory).
21. If you want to change the location of the watch directory after you have set up the channel then select Channels, then highlight the cityEHR Import CDA channel in the list and select Edit Channel from the Channel Tasks menu on the left hand side of the screen.
22. At the top of the window select the Source tab to show a form where you can edit the Directory (i.e. the watched directory) and the Error Move-to Directory.

7. Starting and Stopping Tomcat

As you tinker around with your OHI application you are likely to want to stop and start the Tomcat application server at various times.

This is done by running scripts located in CATALINA_HOME/bin

To start the Tomcat server	startup.bat on Windows systems startup.sh on Mac/Linux/Unix systems
To stop the Tomcat server	shutdown.bat on Windows systems shutdown.sh on Mac/Linux/Unix systems

Since you may need to do this quite often, its a good idea to set up shortcuts to run these scripts.

On Windows you can create a shortcut to startup.bat and shutdown.bat and place them somewhere convenient on your desktop. (Navigate to CATALINA_HOME/bin in Windows Explorer, select startup.bat, right mouse click, select 'Create Shortcut', drag the shortcut onto your desktop).

On the Mac its slightly more complicated since the shell scripts startup.sh and shutdown.sh need to run from inside a (unix) command shell (Mac OS is essentially a linux operating system). You can create executable applications to do this using the Automator tool which is located in your Applications directory.

Run the Automator, create a new script to run the shell script (search for shell in the search box) with the startup.sh script and then save as an Application (which can then be dragged to your Dock for executing easily). Repeat in a similar way for the shutdown.sh script.

See the screen shot below for a summary of how this is done.

