

## Table of Contents

1.Introduction.....	3
2.Open Source Platform for cityEHR.....	4
2.1.Apache Tomcat .....	4
2.2.Orbeon Forms.....	6
3.Integration Engine.....	9
3.1.Mirth Connect.....	9
4.Other Applications Used in Workshop Exercises.....	12
4.1.Open Office.....	12
4.2.Firefox Web Browser.....	12
4.3.Protégé 4.1.....	13
4.4.Serna Free XML Editor.....	14
4.5.Komodo Editor.....	14
5.Configuring the Applications.....	15
5.1.cityEHR Application.....	15
5.2.Mirth Integration Engine.....	16
6.Starting and Stopping Tomcat.....	19

## Revision History

Date	Version	Changes	Owner
14-Dec-2010	1.0	First Version	JC
14-Feb-2011	2.0	Version for release at City University	JC
21-Feb-2011	2.1	Version for general release	JC
22-Feb-2011	2.2	Updated with Firefox installation instructions	JC
28-Feb-2011	2.3	Updated with Komodo Edit installation instructions	JC
01-Oct-2011	2.4	Revised with latest versions of software	JC
26-Oct-2011	2.5	Revised to Orbeon 3.9, new cityEHR deployment	JC
07-Nov-2011	2.6	New version of cityEHR (Sourceforge)	JC
09-Jan-2012	3.0	Updated for new configuration of cityEHR	JC
21-Mar-2012	3.1	Updated for installation of Mirth	JC
01-Nov-2012	3.2	Clarified Orbeon instructions, including Fusion Charts	JC
08-Feb-2013	3.3	Corrected some typos	JC
20-Sep-2013	3.4	Reorganised to clarify for standalone cityEHR installs	JC
03-Oct-2013	3.4.1	Added remote server configuration for cityEHR	JC
12-Jan-2014	3.4.2	Minor updates, including JRE install instructions	JC
01-Dec-2014	3.4.3	Removed requirement for Fusion Charts	JC

---

## 1. Introduction

This document describes the installation of the open source applications required to run the Open Health Informatics workshop.

The following applications are required to run the cityEHR open source health records system. Note that this system is provided at the workshop on a USB stick, ready to run, and so it is only necessary to make the installation below if you want to install and run independently.

- Apache Tomcat (Tomcat 7, but older Tomcat 5.5 or Tomcat 6 are alternatives)
- Orbeon Forms
- Exist native XML database (part of the Orbeon installation)
- Mirth Connect integration/messaging engine (only required for integration services)
- Firefox web browser (any other modern browser can be used, but Firefox is preferred)

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

- Open Office (required)

To install the cityEHR application under Tomcat and Orbeon you require the additional application materials described in Section 5.1

The following applications are required only to complete the workshop exercises - they are not required to run the cityEHR application

- Protégé ontology environment (required)
- Komodo Edit (programmer's file editor - any other equivalent editor can be used)
- Serna Free XML Editor (optional)

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

In addition, some other materials are required to run the workshop exercises and will be provided on the memory stick at the workshop. These materials are usually also available for download after the workshop at [www.openhealthinformatics.org/resources](http://www.openhealthinformatics.org/resources).

---

## 2. Open Source Platform for cityEHR

### 2.1. Apache Tomcat

Tomcat is required to run the Orbeon Forms application framework. If you already have Tomcat installed then you can add the Orbeon application as described below and don't need a fresh installation. If you have a different Java Application server running (e.g. JBoss, Jetty, etc) then Orbeon should run successfully using that. The instructions below are for Tomcat 7, but you can use earlier versions (5.5 or 6) 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](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](http://www.java.com/en/download/help/mac_install.xml)

Tomcat can be downloaded from <http://tomcat.apache.org/download-70>.

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.

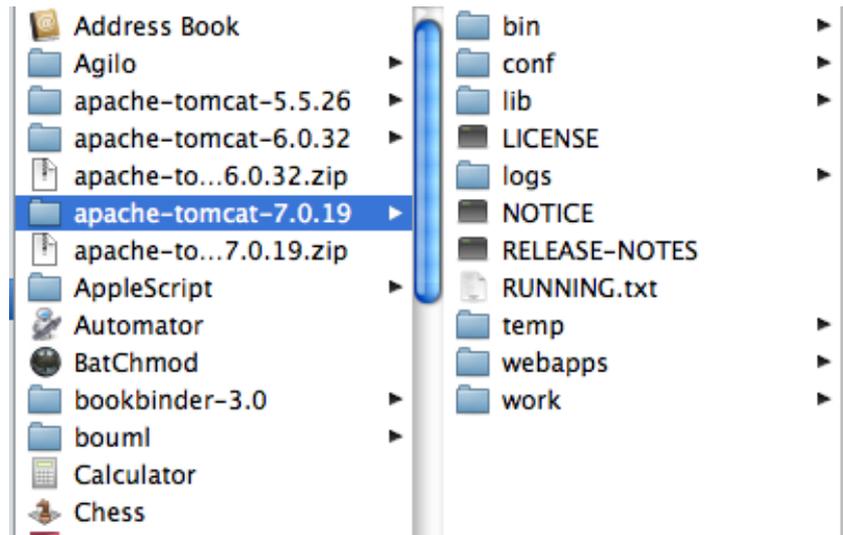
If you are running Windows you may instead run the service installer which is a wizard that will guide you through the installation of Tomcat as a Windows service.

To avoid grief on Windows 7 its best to install outside the c:\Program Files path, unless you are sure you have your Admin permissions under control.

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 5.5 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 6.

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).

# Apache Tomcat/7.0.19



If you're seeing this, you've successfully installed Tomcat. Congratulations!



**Recommended Reading:**  
[Security Considerations HOW-TO](#)  
[Manager Application HOW-TO](#)  
[Clustering/Session Replication HOW-TO](#)

- [Server Status](#)
- [Manager App](#)
- [Host Manager](#)

## Developer Quick Start

- [Tomcat Setup](#)
- [Realms & AAA](#)
- [Servlet Examples](#)
- [Servlet Specifications](#)
- [First Web Application](#)
- [JDBC DataSources](#)
- [JSP Examples](#)
- [Tomcat Versions](#)

### Managing Tomcat

For security, access to the [manager webapp](#) is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 7.0 access to the manager application is split between different users.

[Read more...](#)

[Release Notes](#)

[Changelog](#)

[Migration Guide](#)

[Security Updates](#)

### Documentation

[Tomcat 7.0 Documentation](#)

[Tomcat 7.0 Configuration](#)

[Tomcat Wiki](#)

Find additional important configuration information in:

```
$CATALINA_HOME/RUNNING.txt
```

Developers may be interested in:

[Tomcat 7.0 Bug Database](#)

[Tomcat 7.0 JavaDocs](#)

[Tomcat 7.0 SVN Repository](#)

[Tomcat 7.0 Examples](#)

### Getting Help

[FAQ](#)

[Mailing Lists](#)

The following mailing lists are available:

[announce@tomcat.apache.org](mailto:announce@tomcat.apache.org)  
 Important announcements, releases, security vulnerability notifications. (Low volume).

[users@tomcat.apache.org](mailto:users@tomcat.apache.org)  
 User support and discussion

[taglibs-user@tomcat.apache.org](mailto:taglibs-user@tomcat.apache.org)  
 User support and discussion for [Apache Taglibs](#)

[dev@tomcat.apache.org](mailto:dev@tomcat.apache.org)  
 Development mailing list, including commit messages

If you have problems running the Orbeon application 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
```

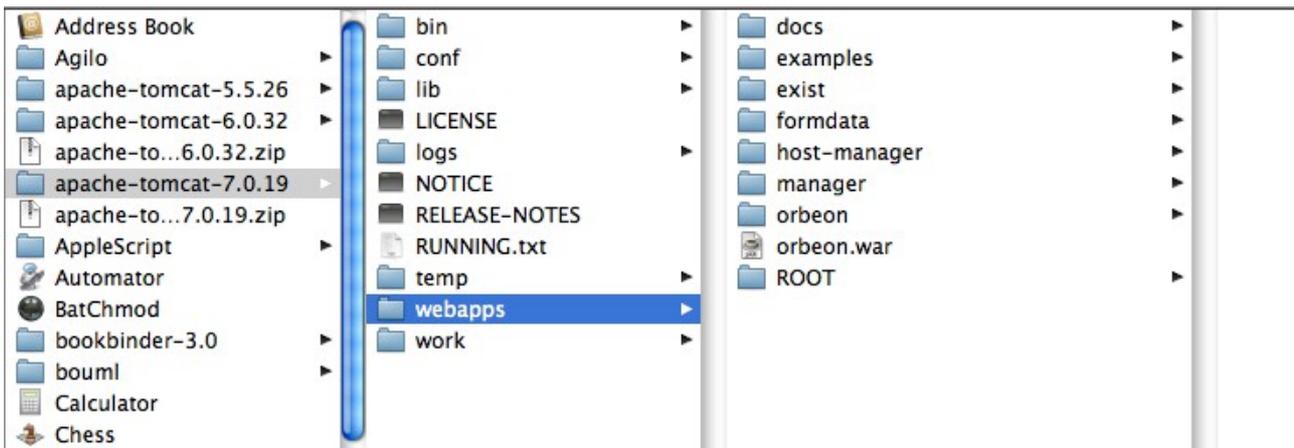
## 2.2. Orbeon Forms

Orbeon Forms can be downloaded from <http://www.orbeon.com/forms/download>. Note that cityEHR currently runs with version 3.9, not the most recent 4.x version of Orbeon.

We will be using the free Community Edition, so you should select the option Download Orbeon Forms 3.9.0 CE in the table which takes you straight to the download on the Orbeon.3.9.0 zip file.

Unzip the Orbeon files and place the orbeon.war file in the CATALINA\_HOME/webapps of your Tomcat installation (where CATALINA\_HOME is the path to where you have Tomcat installed).

Orbeon should then deploy automatically the next time you start the Tomcat server, so you should stop and start Tomcat again to make sure that Orbeon deploys. Once it is you should have a folder structure similar to the one below (screen shot from a Mac on Tomcat 7).



There is one change necessary to the standard installation, to enable webdav browsing of the built-in eXist database. Using a text or programmers file editor, open the file web.xml in the orbeon/WEB-INF folder. Then uncomment the following lines in the file:

```

<!-- Uncomment this for the eXist WebDAV support
  <servlet-mapping>
    <servlet-name>exist-webdav-servlet</servlet-name>
    <url-pattern>/exist/webdav/*</url-pattern>
  </servlet-mapping>
-->

```

Comments in XML are delimited by `<!--` and `-->` so to uncomment the lines, move then trailing `-->` to the end of the first line. leaving you with:

```

<!-- Uncomment this for the eXist WebDAV support -->
  <servlet-mapping>
    <servlet-name>exist-webdav-servlet</servlet-name>
    <url-pattern>/exist/webdav/*</url-pattern>
  </servlet-mapping>

```

There another other change to the Orbeon configuration that is necessary to ensure that dates are shown in European format (i.e. Day-Month-Year). In the folder `orbeon/WEB-INF/resources/config`

Copy the file `properties-local.xml.template` to `properties-local.xml`

Open your `properties-local.xml` file and add the following line:

```

<property as="xs:string" name="oxf.xforms.format.input.date"
  value="[D]/[M]/[Y]"/>

```

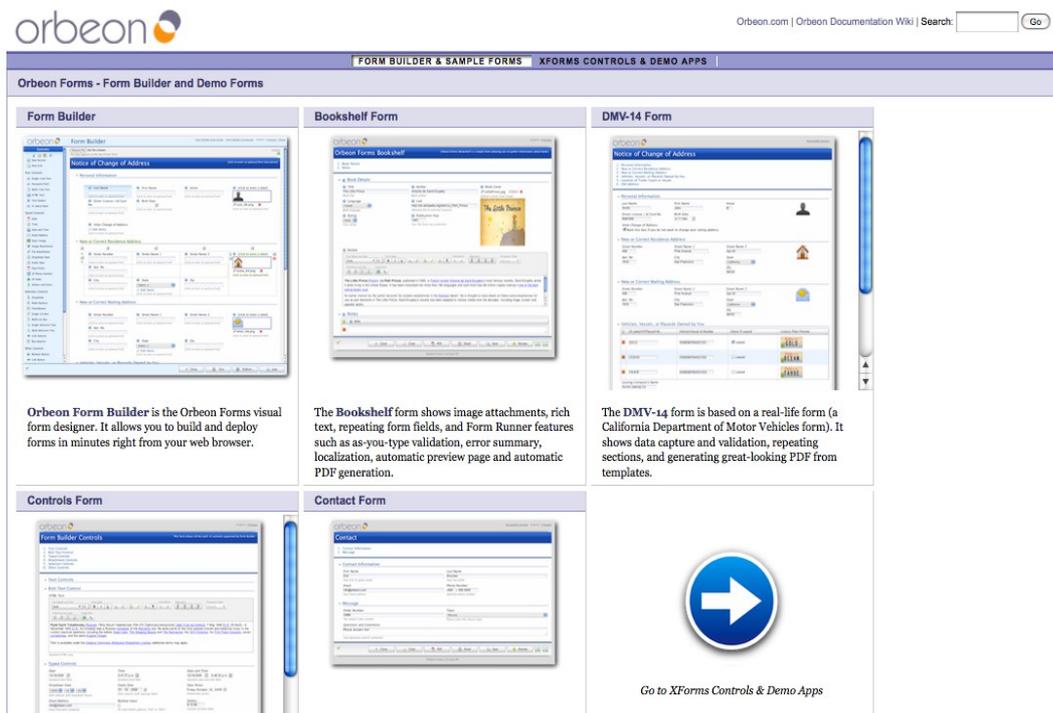
Or if you want to display in the format DD-MMM-YYYY (which is NHS standard) then make the line:

```

<property as="xs:string" name="oxf.xforms.format.input.date"
  value="[D01]-[MNn,*-3]-[Y0001]"/>

```

To check out your Orbeon installation go to <http://localhost:8080/orbeon> in your web browser (or <http://ipaddress:8080/orbeon> where ipaddress is the IP address of your remote server) and you should see a screen similar to the one shown below.



The screenshot displays the Orbeon Forms web application interface. At the top, the Orbeon logo is on the left, and the text "Orbeon.com | Orbeon Documentation Wiki | Search: [input] Go" is on the right. Below the logo, there are two tabs: "FORM BUILDER & SAMPLE FORMS" (selected) and "XFORMS CONTROLS & DEMO APPS". The main content area is titled "Orbeon Forms - Form Builder and Demo Forms" and is divided into several sections:

- Form Builder:** Shows a "Form Builder" interface with a "Notice of Change of Address" form being designed. Below it, a text box states: "Orbeon Form Builder is the Orbeon Forms visual form designer. It allows you to build and deploy forms in minutes right from your web browser."
- Bookshelf Form:** Shows a "Bookshelf" form with image attachments and rich text. Below it, a text box states: "The Bookshelf form shows image attachments, rich text, repeating form fields, and Form Runner features such as as-you-type validation, error summary, localization, automatic preview page and automatic PDF generation."
- DMV-14 Form:** Shows a "DMV-14" form, which is a real-life form from the California Department of Motor Vehicles. Below it, a text box states: "The DMV-14 form is based on a real-life form (a California Department of Motor Vehicles form). It shows data capture and validation, repeating sections, and generating great-looking PDF from templates."
- Controls Form:** Shows a "Form Builder Controls" interface. Below it, a text box states: "Go to XForms Controls & Demo Apps" with a blue circular arrow icon pointing right.

Further configuration of Orbeon required to run cityEHR is described in Section 5.1.

---

## 3. Integration Engine

### 3.1. 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
```

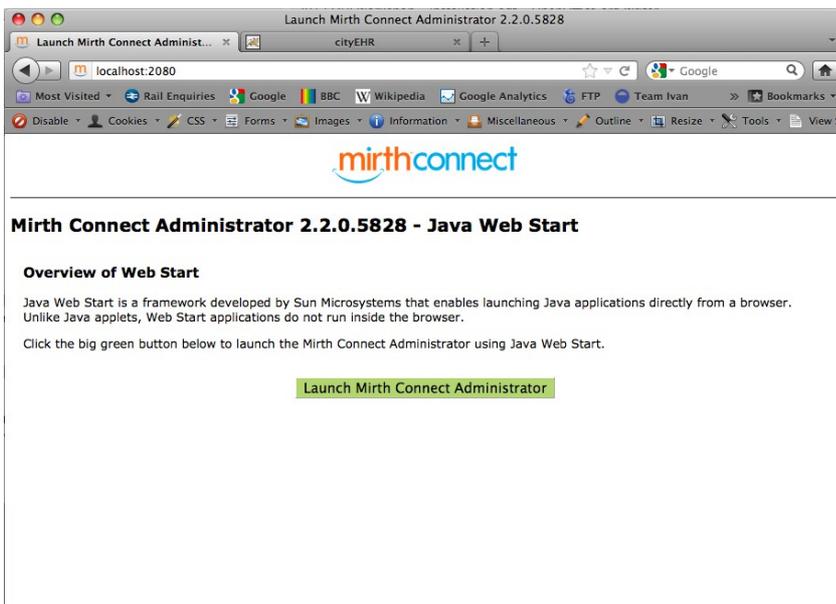
On a Mac you can automate the running of these commands as described in Section 6.

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:

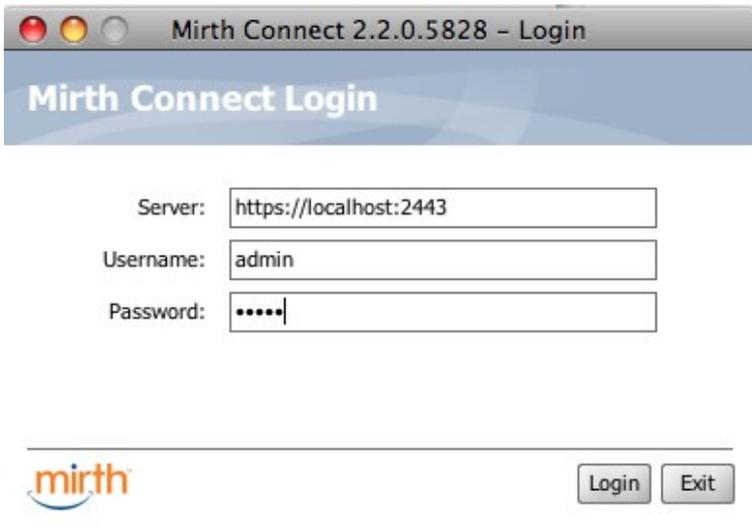


With the Mirth Server running (see above) you should be able to run the Administrator console by clicking the button in the Server Manager. Otherwise (on other platforms, or if the button is not available) you can launch the Administration Console directly from the server by accessing it in a web browser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

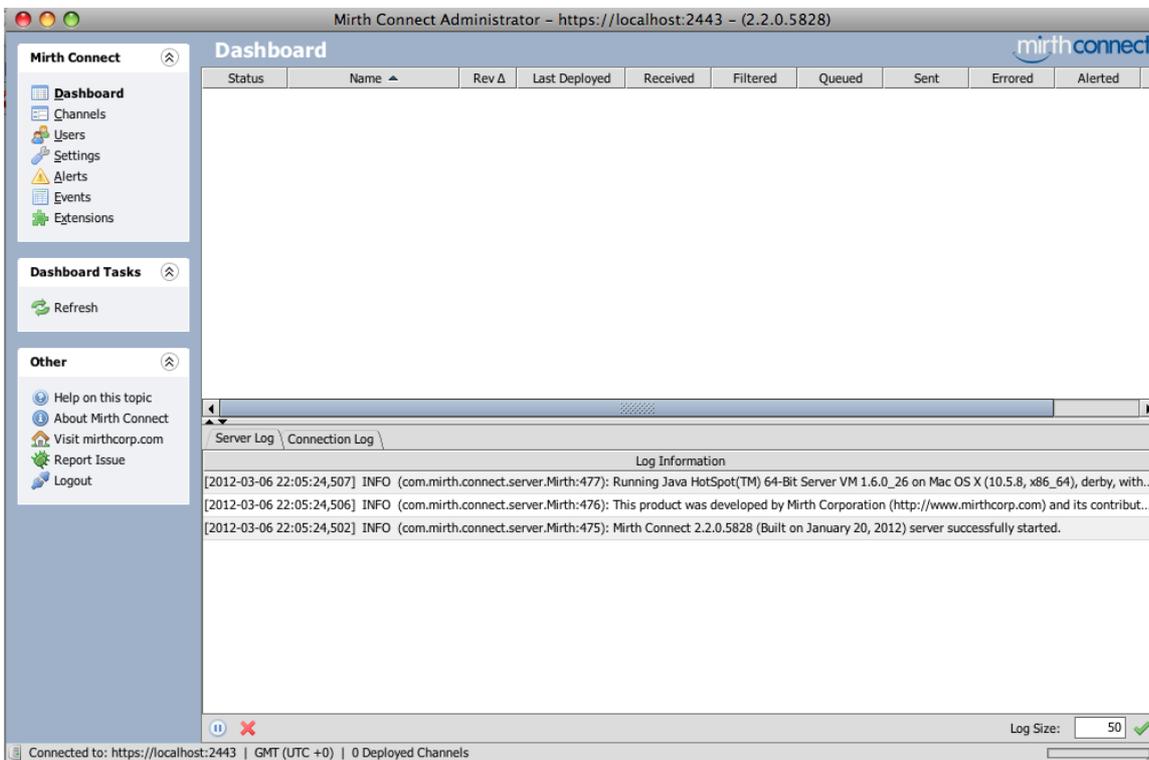


This will save the Java web start for Mirth which you can save in a location of your choice and then use in the future directly by double clicking on it.

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 5.2.

---

## 4. Other Applications Used in Workshop Exercises

### 4.1. 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).

### 4.2. 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/bfbameneiokkgbdmiekhjnfmfcnlidhnm>

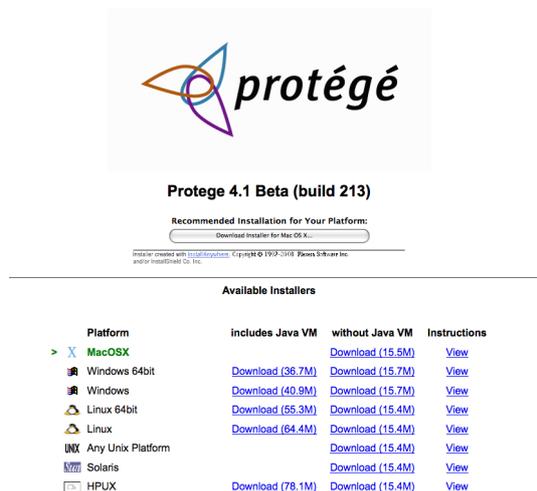
For convenience, we are going to use relative paths to locate files for stylesheets. This requires resetting a security option in Firefox, as follows.

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)

### 4.3. 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



## Welcome to Protégé

- [Create new OWL ontology](#)
- [Open OWL ontology](#)
- [Open OWL ontology from URI](#)
- [Open from the TONES repository](#)

### Open recent

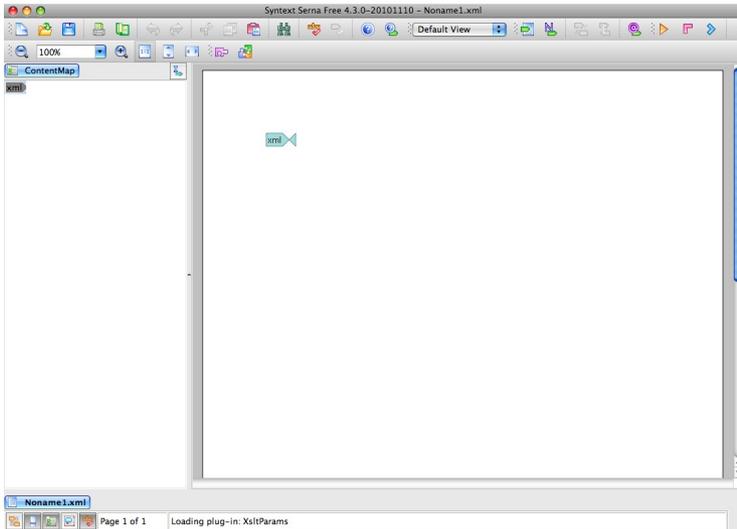
[/Users/johnchelsom/Documents/city university/Projects/Nottingham Project/ORCHID Coding Scheme/owl/orchid.owl](#)  
[/Users/johnchelsom/Documents/eleven/projects/P1816 - XML Summer School/Hands-on Introduction/2010/Exercises/E](#)  
[/Users/johnchelsom/ontologies/Ontology1268402456791/Ontology1268402456791.owl](#)  
[/Users/johnchelsom/Documents/city university/2009-10 Lectures/INM358- Knowledge Management in Healthcare /Ont](#)

---

## 4.4. Serna Free XML Editor

Serna Free XML editor can be downloaded from <http://www.syntext.com/downloads/serna-free/>

Select the correct download for your platform and follow the installation instructions provided. On successful installation you should be able to start up Serna Free and select Document | New Document from the main menu. Click OK to accept the Default category and enter a name for the root element (e.g. xml). You should then see a screen similar to the one below.



## 4.5. 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.

## 5. Configuring the Applications

### 5.1. cityEHR Application

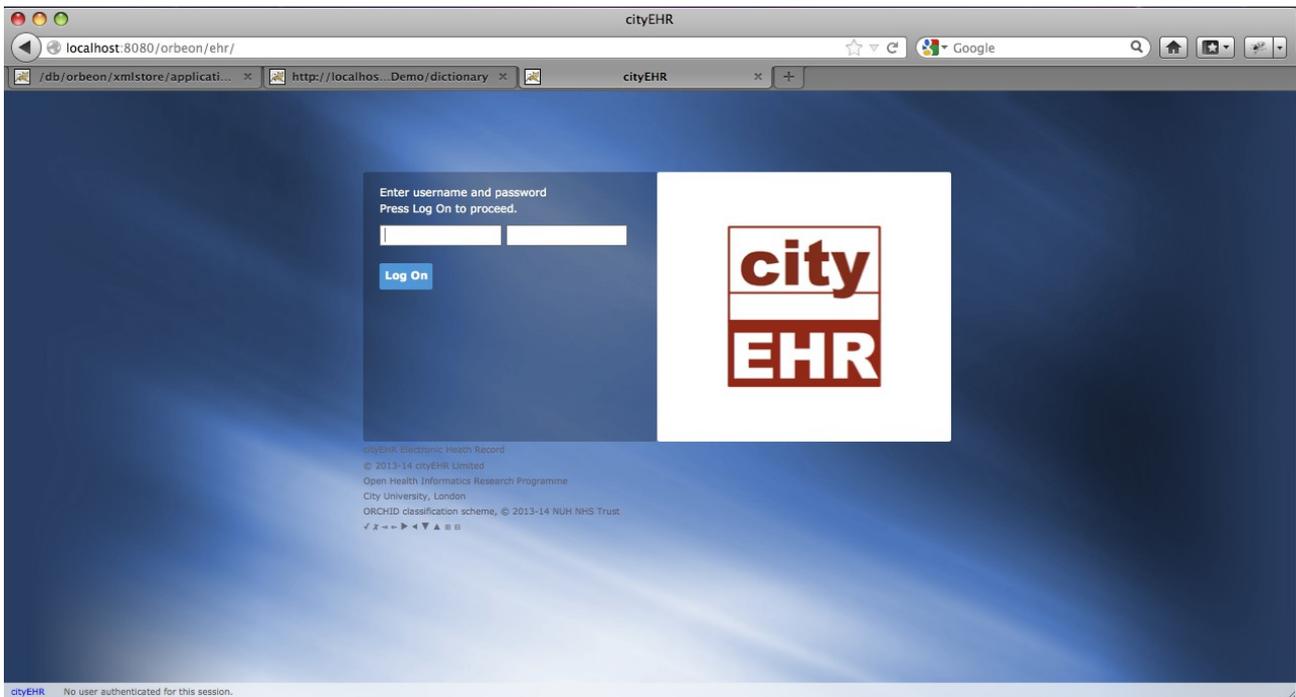
The latest version of cityEHR can be downloaded from <https://sourceforge.net/projects/cityehr/files>

The complete cityEHR is in the zip file ehr.zip – download and unzip to create a folder called ehr.

Move (or copy) the directory called ehr to the Tomcat Orbeon apps directory:

CATALINA\_HOME/webapps/orbeon/WEB-INF/resources/apps

If running cityEHR on your local machine you should now be able to open your web browser and go to <http://localhost:8080/orbeon/ehr/> to see the log on screen of cityEHR similar to the one shown below. If the page is not found, check that you have included the trailing slash '/' in the web address.



If you are running on a remote server then you will need to change some of the configuration settings for cityEHR. These are in a file called view-parameters.xml inside the ehr folder at CATALINA\_HOME/webapps/orbeon/WEB-INF/resources/apps.

Ideally you should edit view-parameters.xml using an XML aware editor such as the Komodo editor listed above or in an XML editing environment such as oXygen (see <http://www.oxygen.com> - this is not open source, but is available cheaply for academic users). Once you have made the changes it is important that view-parameters.xml is still a well formed XML document. If not, it will not load up when cityEHR is started and the application will not run. You can check whether view-parameters.xml is well formed by opening it in a web browser (use the File|Open menu command) - if there are any problems then the browser will display an error message from its XML parser which should direct you to where the problem is.

---

To configure cityEHR running on a server at (for example) IP address 192.168.1.1 you need to make the following changes in view-parameters.xml

`<staticFileRoot>http://localhost:8080/orbeon/ehr/resources</staticFileRoot>`

changes to

`<staticFileRoot>http://192.168.1.1:8080/orbeon/ehr/resources</staticFileRoot>`

`<databaseLocation>http://admin:@localhost:8080/orbeon/exist/rest</databaseLocation>`

changes to

`<databaseLocation>http://admin:@192.168.1.1:8080/orbeon/exist/rest</databaseLocation>`

`<mediaDatabaseLocation>/exist/rest</mediaDatabaseLocation>`

changes to

`<mediaDatabaseLocation>http://admin:@192.168.1.1:8080/orbeon/exist/rest</mediaDatabaseLocation>`

`<auditlogDatabaseLocation>/exist/rest</auditlogDatabaseLocation>`

changes to

`<auditlogDatabaseLocation>http://admin:@192.168.1.1:8080/orbeon/exist/rest</auditlogDatabaseLocation>`

Once these changes have been made you should be able to access the cityEHR log on page at <http://192.168.1.1:8080/orbeon/ehr/> (or equivalent for your server IP address)

## 5.2. 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 Orbeon forms 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",
"xml:db:exist://admin:@localhost:8080/orbeon/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.

## 6. 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 folder.

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.

