

---

**PatientOS**

---

**PatientOS  
Software  
Developer Guide**

**Version 0.94.001**

Software	Last printed 7/15/2009 2:14:00 PM
Developer Guide	Version 0.94.001

## Revision History

<b>Date</b>	<b>Version</b>	<b>Description</b>	<b>Author</b>
04/11/2009	0.92	Initial	Greg Caulton
07/08/2009	0.94.001	Add Source Control Environment	William Facey

Software	Last printed 7/15/2009 2:14:00 PM
Developer Guide	Version 0.94.001

# Contents

- 1. OVERVIEW.....3**
  - 1.1 DEFAULT SETTINGS .....3
  - 1.2 DEFINITIONS .....3
- 2. DEVELOPMENT ENVIRONMENT .....4**
  - 2.1 JAVA VERSION 1.6.....4
  - 2.2 ECLIPSE IDE .....4
  - 2.3 PATIENTOS SOURCE.....4
  - 2.4 ECLIPSE (OPTIONAL).....5
  - 2.5 DEVELOPMENT DATABASE.....9
  - 2.6 BUILDING PATIENTOS ON LINUX .....11
  - 2.7 PATIENTOS SOURCE CONTROL .....12
- CUSTOM CONTROLLER VIEWS.....17**
  - 2.8 CREATING PLUGIN VIEWS .....17
- 3. SOFTWARE DEPLOYMENT.....18**
  - 3.1 DEVELOPMENT DATABASE.....18
  - 3.2 UPDATE HELP FILE .....18
  - 3.3 BUILD SOFTWARE.....19
  - 3.4 CREATE UPGRADE .....19
  - 3.5 CREATE DEPLOYMENT.....20
  - 3.6 CREATE WINDOWS INSTALLER.....21
- 4. CLEANUP SQL.....22**
  - CREATE LINUX INSTALL.....23
- 5. DIAGNOSTIC TOOL.....23**



Software	Last printed 7/15/2009 2:14:00 PM
Developer Guide	Version 0.94.001

## 2. Development Environment

### 2.1 Java version 1.6

Java version 1.6 is a requirement to build and run the software. Download latest version of the JDK from <http://java.sun.com/javase/downloads/index.jsp>

**JDK 6 Update 5** » Download

The Java SE Development Kit (JDK) includes the Java Runtime Environment (JRE) and command-line development tools that are useful for developing applets and applications.

### 2.2 Eclipse IDE

Eclipse is one of many Java IDE's that can be used to modify the source code. Download the latest version of Eclipse from <http://www.eclipse.org/downloads/>

#### Eclipse Europa Winter Maintenance Packages - Windows (compare packages)



**Eclipse IDE for Java Developers** - Windows (79 MB)

The essential tools for any Java developer, including a Java IDE, a CVS client, XML Editor and Mylyn. [Find out more...](#)

Windows  
Linux  
Mac OS X

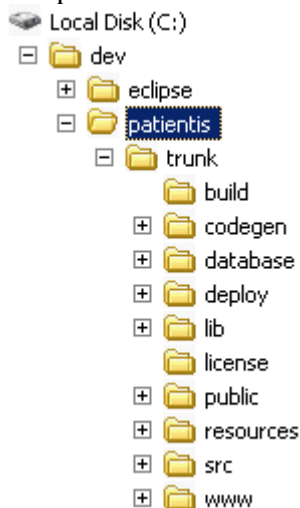
Unzip Eclipse into a directory e.g. c:\dev or /usr/local/eclipse

### 2.3 PatientOS Source

Download the PatientOS source code for the latest release. [http://sourceforge.net/project/showfiles.php?group\\_id=201006](http://sourceforge.net/project/showfiles.php?group_id=201006)

patientos		
Latest	v0.60	(2008-03-14 11:46)
	patientos-0.60-documentation.zip	965771
	PatientOS-0.60-Setup.exe	248531087
	patientos-0.60-setup.tar.gz	256569998
	patientos-060-src.zip	85012949

Unzip the contents



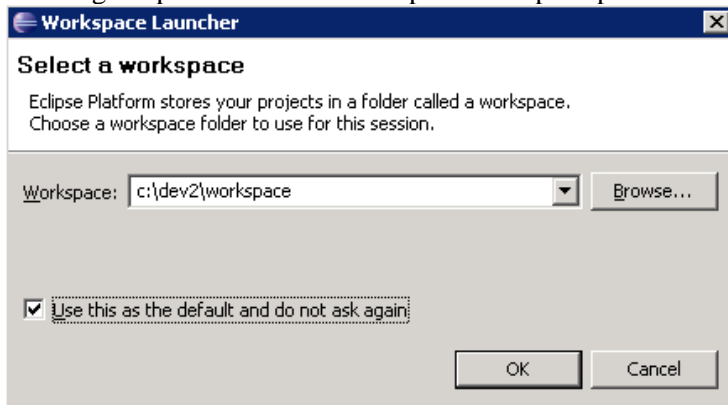
Software	Last printed 7/15/2009 2:14:00 PM
Developer Guide	Version 0.94.001

## 2.4 Eclipse (Optional)

Eclipse is not required to build the code but the instructions are included here for convenience.

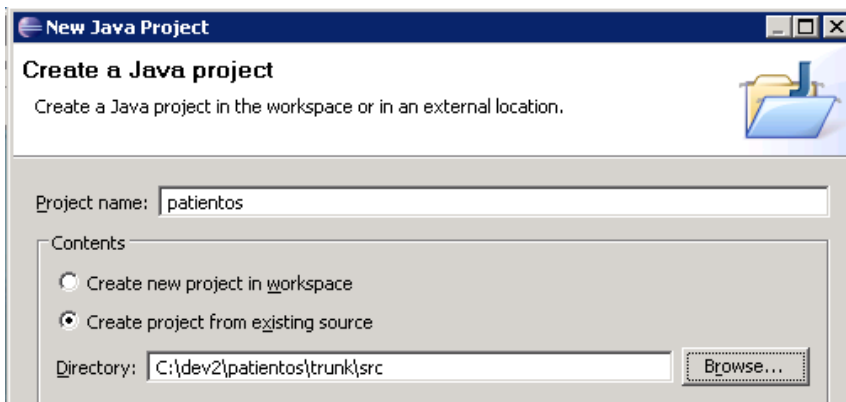
### 2.4.1 Start Eclipse

Running eclipse.exe will start Eclipse with a prompt to choose a workspace enter c:\dev\workspace for example



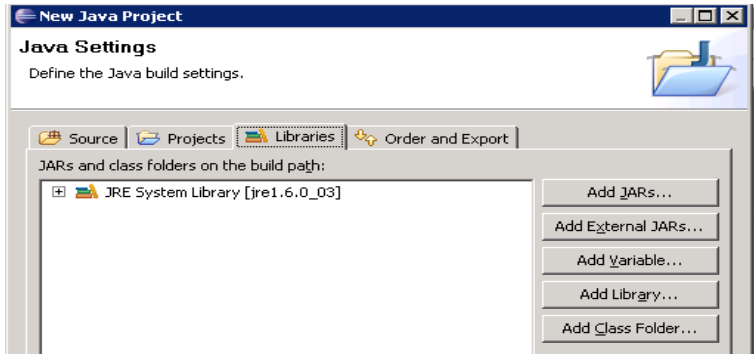
### 2.4.2 New Java Project

Use menu File → New → Java Project selecting to build from existing source



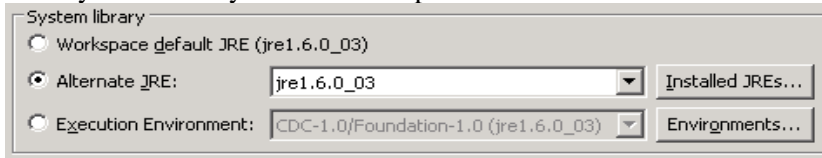
Software	Last printed 7/15/2009 2:14:00 PM
Developer Guide	Version 0.94.001

Press Next and select the libraries tab. Note the System Library.

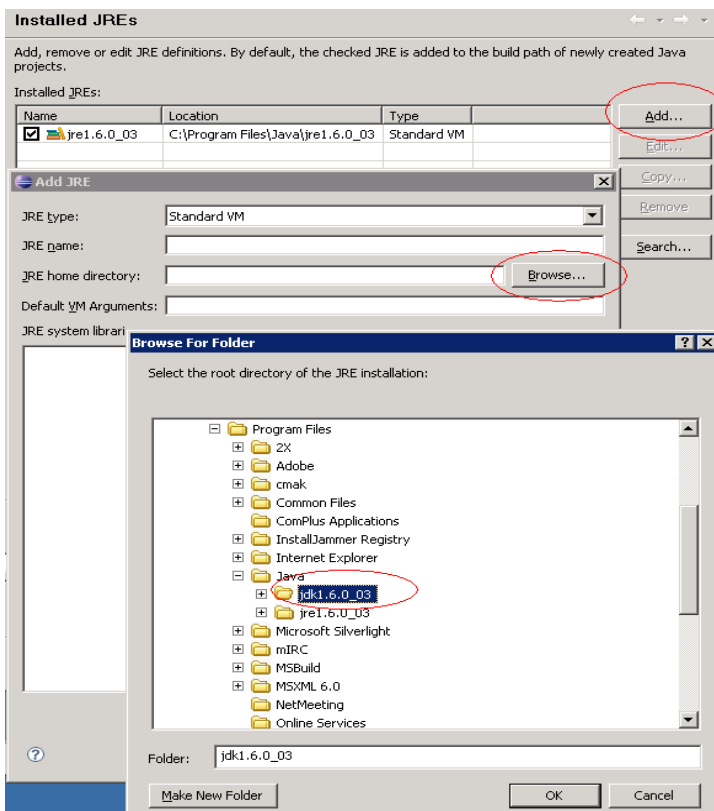


The System Library needs to be the JDK1.6.0\_x installed not the JRE. If this is not the case select and remove this library and press the Add Library button.

JRE System Library will default – press Next and choose the Alternate JRE radio button



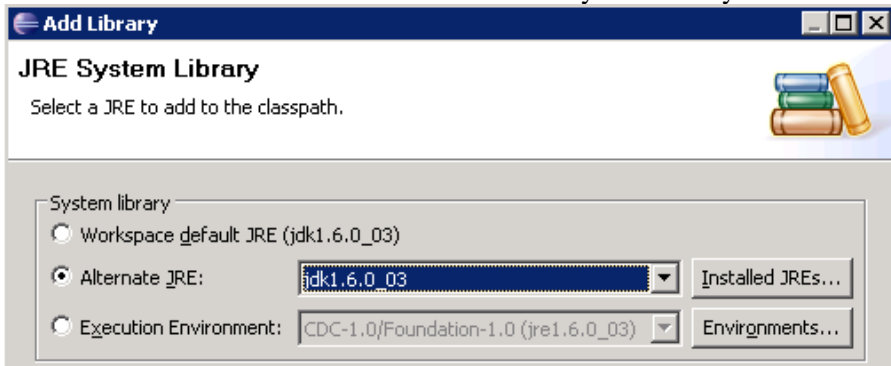
If the jdk is not installed press Add and then Browse to select C:\Program Files\Java\jdk1.6.0\_03 or equivalent in Linux.



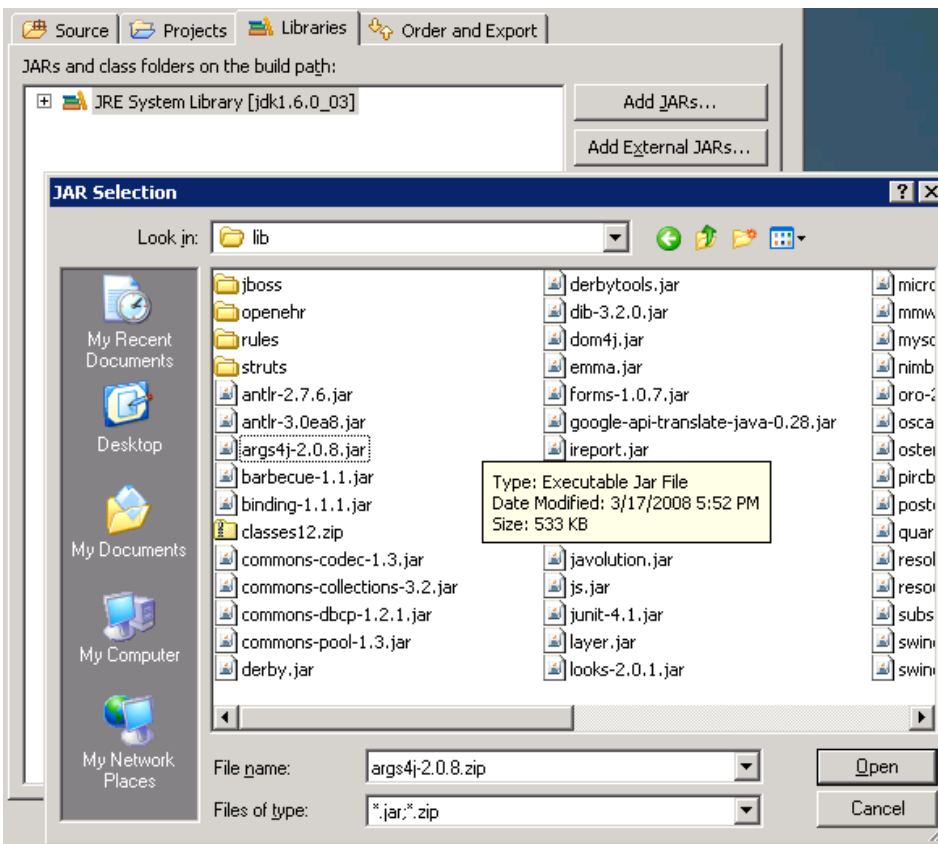
Software	Last printed 7/15/2009 2:14:00 PM
Developer Guide	Version 0.94.001

### 2.4.3 PatientOS Libraries

You should now have the JDK 1.6 selected as a system library.



Back on the Libraries tab press External Jars and browse and select all the libraries in C:\dev\patientos\trunk\lib as well as each subdirectory – jboss, rules and struts.



Click finish. Close the Welcome page, task list and outline.

Software	Last printed 7/15/2009 2:14:00 PM
Developer Guide	Version 0.94.001

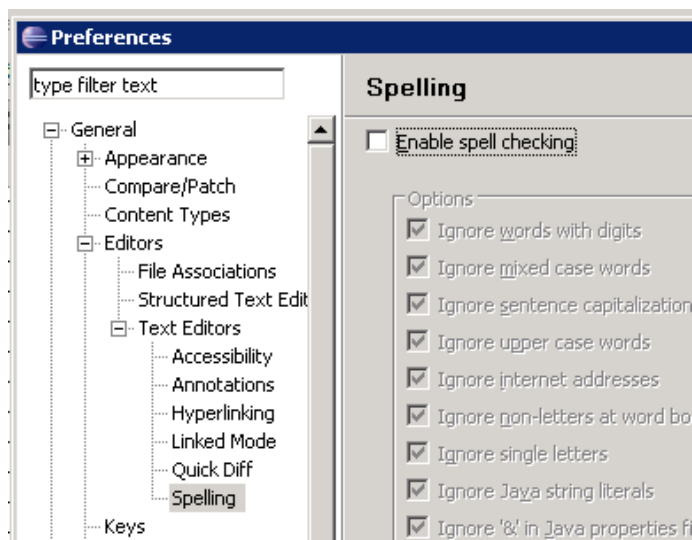
#### 2.4.4 Compilation

The 0.60 src zip includes files that should have been deleted from the subversion repository. Delete the following:

`com.patientis.business.calendar.AppointmentList.java`  
`com.patientis.pluginclient.scheduling.AppointmentList` (entire package)  
`com.unittest.openehr` (entire package)  
`com.patientis.client.clinical.RecordItemUtility.java`  
`com.patientis.framework.itext.DataColumnAdapter.java`  
 Right click and refactor `com.patientis.client.external.ImportCsv` to be `com.patientis.client.external.ImportCSV` (all caps)

If none of the Jars were missed then the system should compile with 0 errors and and several hundered minor warnings.

Disable spell check using menu Window → Preferences → General—Editors—Text Editors—Spelling



Software	Last printed 7/15/2009 2:14:00 PM
Developer Guide	Version 0.94.001

## 2.5 Development Database

In order to effectively develop new applications a

- a) Dedicated development database is required to add system level content.
- b) Dedicated QA database is required to test against.

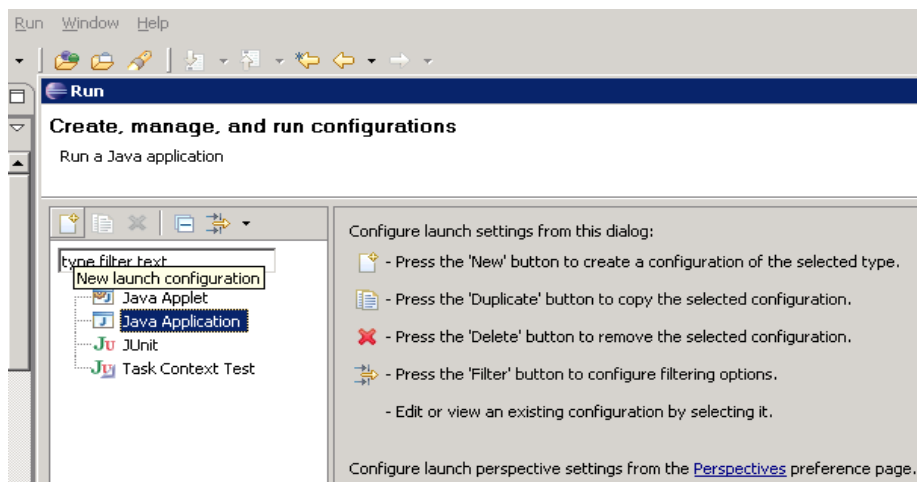
For now install the full version of PatientOS into a directory e.g. C:\program files\patientos\0.60 or /usr/local/patientos/060

Start the application server and use that server to connect to. Any local changes can be tested by running the PatientOS client from Eclipse. Any server side changes need to be compiled using the ant build script and deployed to the C:\program files\patientos\0.60\server\appserver\server\default\deploy directory (or equivalent on Linux).

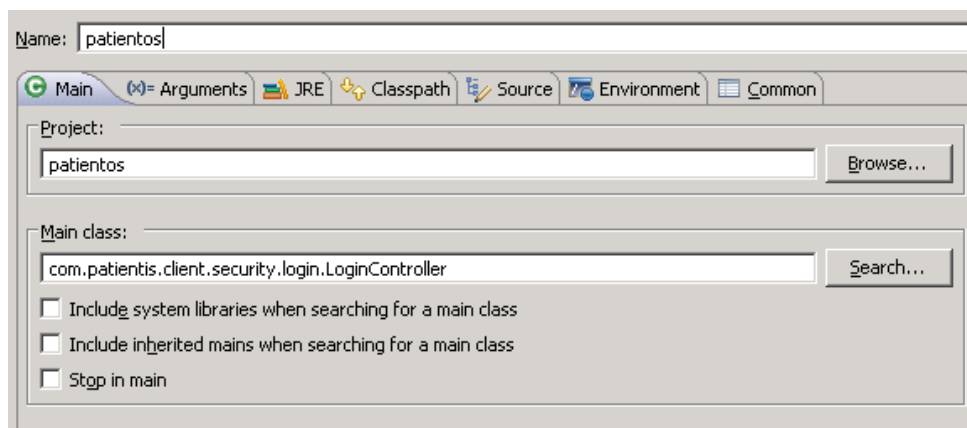
### 2.5.1 Starting the Client in Eclipse

Eclipse requires the equivalent properties found in the client\bin\patientos.bat file to start.

Use menu Run → Open Rune Dialog and select Java Application and location the New launch configuration toolbar button.



Enter com.patientis.client.security.login.LoginController as the main class with patientos for the configuration name.

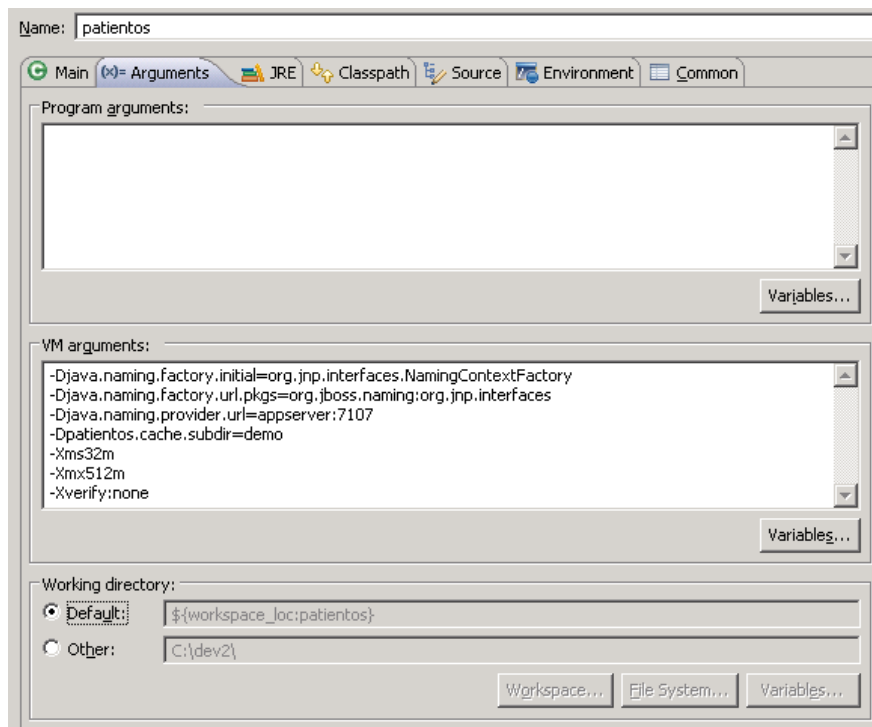


Software	Last printed 7/15/2009 2:14:00 PM
Developer Guide	Version 0.94.001

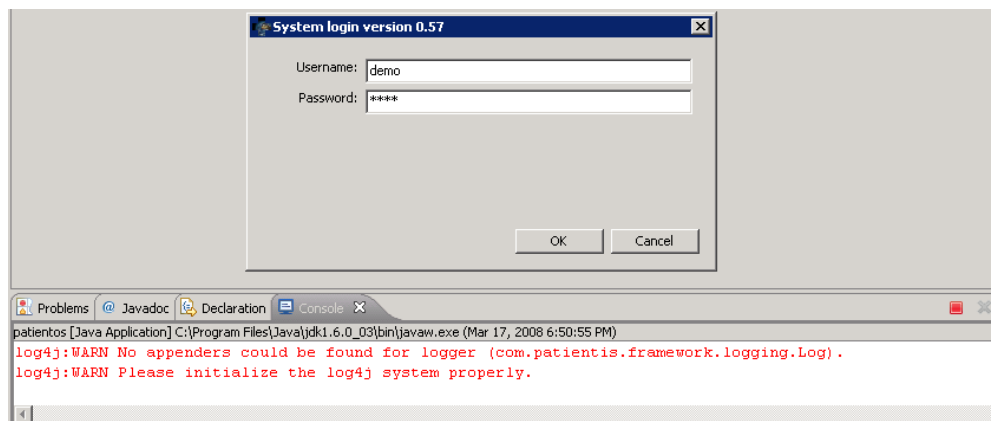
On the arguments tab the following VM properties are needed

```
-Djava.naming.factory.initial=org.jnp.interfaces.NamingContextFactory
-Djava.naming.factory.url.pkgs=org.jboss.naming:org.jnp.interfaces
-Djava.naming.provider.url=localhost:7107
-Dpatientos.cache.subdir=demo
-Xms32m
-Xmx512m
-Xverify:none
```

Note the working directory if set will have a PatientOS subdirectory created with the local cache.



Note localhost can be changed to the server where the application server is running if different from the local machine.



Software	Last printed 7/15/2009 2:14:00 PM
Developer Guide	Version 0.94.001

## 2.6 Building PatientOS on Linux

### 2.6.1 Subversion

An alternative to downloading the last release of PatientOS is to download the source using subversion. Download the subversion client from [http://subversion.tigris.org/project\\_packages.html](http://subversion.tigris.org/project_packages.html) and check out the code with the command

```
svn co https://patientos.svn.sourceforge.net/svnroot/patientos patientos
```

You should have Java version 1.6 installed. Validate with the command `java -version`  
Set JAVA\_HOME to your 1.6 directory e.g.

```
JAVA_HOME=/usr/local/java/1.6xxx  
export JAVA_HOME
```

Download ANT version 1.7 (required to compile) from <http://ant.apache.org/bindownload.cgi>  
Set ANT\_HOME to your 1.7 directory e.g.

```
ANT_HOME=/usr/local/apache-ant-1.7.0  
export ANT_HOME  
PATH=$ANT_HOME/bin:$PATH
```

Validate ANT 1.7 is being used with the command `ant -version`

Increase memory to compile the source

```
ANT_OPTS=" -Xmx1024m"  
export ANT_OPTS
```

Change directory into the build directory containing build.xml and run

```
ant
```

Software	Last printed 7/15/2009 2:14:00 PM
Developer Guide	Version 0.94.001

### 2.6.2 Deploy

```
patientos/client/patientis.jar
patientos/client/resources.jar
patientos/deploy/patientis.ejb3
patientos/deploy/patientis.jar
```

These four files are the typical files which change from upgrade to upgrade along with the database schema and contents.

### 2.6.3 Troubleshooting

On some hosted servers you may need to resolve and replace the host name with the IP e.g.

```
svn co https://66.35.250.146/svnroot/patientos patientos
```

Other issues come from not having JAVA\_HOME, ANT\_HOME or ANT\_OPTS set appropriately.  
“The system is out of resources “ or “Exception in thread "main" java.lang.NoClassDefFoundError”

ant --noconfig will resolve the latter error.

### 2.6.4 Committing changes

```
svn commit --username caultonpos -m "v0.92" patientos
```

## 2.7 PatientOS Source Control

### 2.7.1 Subversion Server

The PatientOS source code repository utilizes Subversion 1.5.1 (r32289) and is hosted by SlikSVN at:

```
https://svn1.sliksvn.com/patientos
```

All administration tasks are performed by the support group ([support@sliksvn.com](mailto:support@sliksvn.com)) at SlikSVN.

The repository can be browsed with any internet browser and can be modified with any of the supported SVN clients (Subversion client, TortoiseSVN, Subclipse, etc.).

### 2.7.2 PatientOS Repository Structure

Any SVN client can be used to create folders in the repository. The PatientOS structure must adhere to the following guideline:

```
/trunk
```

(All folders to be kept under version control)

```
/branches
```

Software	Last printed 7/15/2009 2:14:00 PM
Developer Guide	Version 0.94.001

(All Release, Developer, and feature branches created.)

/tags

(All tags created to give a more meaningful name to a revision number.)

Folders kept under version control include all files required in the development environment to change, build, and test the code including source code, build procedures, development tools, and database schema & data files. All documentation (manuals, guides, help files) are also included.

The Source code can be found in the following folders:

```

/src
/public      Patient Portal
/www

```

Build procedures require files within the following folders:

```

/build
/lib
/reference
/resources
/icons
/tools

```

Testing requires the database schema and data files to be updated within the folder:

```

/database

```

The branches folder contains all branches and sub-branches created. The branching structure utilized is:

```

/branches
  /v094.001           A Release branch of the trunk.
  /wfacey_v094.001   A Developer branch of the release.
  /POSxxx_v094.001   A Feature branch of the release.
  /POSxxx_wfacey_v094.001 A Feature branch of the developer branch.

```

A Release branch is created directly from the trunk to separate the release work from the main product line. More than one Release branch is created when work on multiple releases must be done in parallel. When this is done, the release branches must be synchronized with the trunk after a release has been merged back (reintegrated) into the trunk.

A Release branch is typically never deleted unless there is no need for the code traceability data captured on the commits.

A Developer branch is created directly from a Release branch to allow developers to commit their changes without fear of impacting other developers until they deem the code ready for integration into the release branch.

A Developer branch is typically never deleted unless there is no need for the code traceability data captured on the commits.

A Feature branch is not typically created but may be required to allow a feature to be implemented for a release in parallel to and independently of the other features of the release. It may be created directly from the release or developer branch. The feature implemented is typically new to the product and resolves a single product request. These branches can be deleted after they have been reintegrated into the release or developer branch.

The tags folder contains all “snapshot” tags created that identify a snapshot in time of the product. I.e. Instead of tracking trunk revision 123 as release 0.94.001, it can be tagged as v094.001 just as a Release branch is. Subversion doesn’t differentiate between a branch and a tag which means the result can be achieved by synchronizing the release branch with the reintegrated trunk release. This could lead to confusion as to whether a release branch is “released” or not. The tags folder clearly identifies those releases ready for deployment.

Software	Last printed 7/15/2009 2:14:00 PM
Developer Guide	Version 0.94.001

Branches and/or Tags are created using the following Subversion client command:

```
svn copy https://svn1.sliksvn.com/patientos/trunk  
https://svn1.sliksvn.com/patientos/branches/<branchName> -m "Release branch  
<branchName> created."
```

OR

```
svn copy https://svn1.sliksvn.com/patientos/trunk -r 123  
https://svn1.sliksvn.com/patientos/tags/<tagName> -m "Tagged Release  
v094.001 as <tagName>."
```

### 2.7.3 Establish a Working Directory

A Working directory linked to the appropriate repository branch is required before any changes can be made to the repository. This can be done in the following ways:

- Checkout existing repository sub-tree into a new working directory.

This is the typical action when a developer creates a new working directory. Simply use the svn checkout command to establish your working directory. The checkout doesn't modify the repository; it only creates a private copy of the repository in the folder specified. If no local folder is specified, the repository folder name is used. If multiple repository sub-trees are specified they can be copied into separate working directories by not specifying a working directory or into a single working directory by specifying the working directory.

```
svn checkout https://svn1.sliksvn.com/patientos/branches/v094.001/src  
https://svn1.sliksvn.com/patientos/branches/v094.001/public  
https://svn1.sliksvn.com/patientos/branches/v094.001/www  
v094.001
```

The example above creates a single working directory "v094.001" with copies of the "src", "public", and "www" folders. Had the working directory "v094.001" not been specified, each folder would have a separate working directory.

- Checkout existing repository sub-tree into an existing un-versioned directory.

The same action is performed as above. The checkout will merge the repository files with any files having the same name residing in the same folder. This means you have to discard all unwanted changes made to the files. Your existing directory is now a working directory for the repository sub-tree.

```
svn checkout (as above)
```

Software	Last printed 7/15/2009 2:14:00 PM
Developer Guide	Version 0.94.001

```
svn revert [local folder] --depth=infinity
```

- Switch an existing working directory to a different repository sub-tree.

This is used when an existing working directory should reflect a different repository sub-tree. The files in the working directory will be updated with the files from the new sub-tree.

```
svn sw[itch] URL [Path]
```

Example: `svn sw`

[https://svn1.sliksvn.com/patientos/branches/wfacey\\_v094.001/src](https://svn1.sliksvn.com/patientos/branches/wfacey_v094.001/src) src

The example above will switch the working directory of my current “src” folder to the “src” folder within my developer branch. Perhaps I had performed the checkout as in the previous example above such that it was a working directory of the “src” folder within the release branch.

#### 2.7.4 Modifying Files

All text files can be simply modified as needed from your working directory. At some point you will commit your changes back to the repository (typically your personal branch) and they will be updated into the previous versions. The commit command only updates the repository and doesn't update the working directory.

```
svn ci || commit [Path]
```

Example: `svn ci src\com\patientis\model`

The example above will commit all changes to the “src\com\patientis\model” sub-tree of the repository.

Any binary files (Word/Excel files) to be modified should be handled with extra care. When the modification is committed to the repository it will replace the previous version which means someone could potentially lose their changes unless we use the locking feature to ensure our reserved files cannot be replaced or locked until we have committed our changes. This means we shouldn't modify binary files from a developer branch as more than one developer may be working on changes in their own branch. We would instead modify them from the release branch where the file is first locked, then modified, and then committed and unlocked. See the example below:

```
svn lock Target...
```

Example: `svn lock`

<https://svn1.sliksvn.com/patientos/v094.001/documentation/SoftwareDeveloperGuide.doc>

The example above locks the specified document within the release branch thus preventing any other user to apply their changes to it. The lock is automatically released when you commit your changes to the branch unless you specify on the

Software	Last printed 7/15/2009 2:14:00 PM
Developer Guide	Version 0.94.001

commit not to release the locks with the `-no-unlock` option. If you have locked a file and decide not to make any changes, you must unlock the file as below:

```
svn unlock Target...
```

Software	Last printed 7/15/2009 2:14:00 PM
Developer Guide	Version 0.94.001

## Custom Controller Views

### 2.8 Creating plugin views

- 1) Create a custom controller  
e.g. `com.patientis.business.controllers.DefaultCustomMessagesController`
- 2) Create a renderer  
e.g. `com.patientis.business.controllers.DefaultCustomMessagesRenderer`
- 3) Create a settings form and settings file  
e.g. `com.patientis.business.settings.DefaultCustomMessageListSettings`
- 4) Customize the controller
  - a. retrieve the settings for the control in *clientInitializeControl*
  - b. make the filter list call
- 5) Add the custom controller in the application referencing the settings form
- 6) Create a toolbar or use an existing system toolbar
- 7) Add the CustomTable control to the form with the custom controller
- 8) Add unique context to the form which should match the toolbar

Software	Last printed 7/15/2009 2:14:00 PM
Developer Guide	Version 0.94.001

### 3. Software Deployment

#### 3.1 Development Database

##### 3.1.1 Update version

Update version in `com.patientis.framework.utility.Software.Software.java`  
update refs set display= '0.92' where reference\_group='SystemVersion' and  
ref\_key='DATABASE' ;

##### 3.1.2 Update version

```
select distinct 'ALTER TABLE ' || tablename || ' ALTER COLUMN system_ref_id SET  
DEFAULT 100092;' as cmd from tablelibrary where columnname = 'system_ref_id'
```

##### 3.1.3 Insert version

```
insert into refs (ref_id, reference_group, ref_key,display)  
values (100092,'SystemVersion','V092','0.92')
```

##### 3.1.4 remove existing directories [on second run]

If this is the second deployment on the same day remove files in these directories

```
C:\dev\local\deploy\yyyymmdd  
C:\dev\patientis\trunk\deploy\upgrade\data\yyyymmdd
```

##### 3.1.5 update software.java

with new version

##### 3.1.6 main generator

```
run com.patientis.upgrade.codegenerator.MainGenerator C:\dev\patientis\trunk\src
```

#### 3.2 Update Help File

```
copy/y C:\dev\patientis\trunk\help\patientos-help.jar C:\dev\patientis\trunk\lib  
copy/y C:\dev\patientis\trunk\documentation\help_admin\sysadmin-help.jar C:\dev\patientis\trunk\lib
```

Software	Last printed 7/15/2009 2:14:00 PM
Developer Guide	Version 0.94.001

### 3.3 Build Software

#### 3.3.1 *buildcode.bat*

Compile the java source files.

```
cd c:\dev\patientis\trunk\build
buildcode.bat
```

#### 3.3.2 *buildportal.bat*

Compile the java source files.

```
cd c:\dev\patientis\trunk\build
buildportal.bat
```

```
cd C:\dev\patientis\trunk\build
backuptest.bat
```

### 3.4 Create Upgrade

Note upgrade should blow away the client cache.

#### 3.4.1 *Rebuild demopos schema with new schema*

```
psql -U demopos demopos drop schema public cascade; create schema public;
psql -U demopos demopos < testis.sql
```

#### 3.4.2 *Reset sequences*

#### **DOUBLE CHECK YOU ARE NOT RUNNING THIS IN DEVELOPMENT**

```
select 'drop sequence ' || tablename || '_s;' as drop
from
(select tablename from tablelibrary where tableind = 1 group by tablename having count(tablename) > 2) x
where not tablename like 'view%'
union
select distinct 'create sequence ' || tablename || '_s start 50000000;' as drop
from (select tablename from tablelibrary where tableind = 1 group by tablename having count(tablename) > 2) y
where not tablename like 'view%'
order by 1 desc
```

#### **DOUBLE CHECK YOU ARE NOT RUNNING THIS IN DEVELOPMENT**

#### 3.4.3 *Run Cleanup*

Run delete sql on demopos

Software	Last printed 7/15/2009 2:14:00 PM
Developer Guide	Version 0.94.001

#### 3.4.4 *Export Database Schema*

```
pg_dump --schema-only -i -h localhost -p 5432 -U postgres -F p -O -v -f "demopos_schema.sql" demopos
```

#### 3.4.5 *Offline Copy*

```
del C:\dev\patientis\trunk\deploy\upgrade\02data\2*

com.patientis.upgrade.dbcopy.OfflineCopy
C:\dev\patientis\trunk\src\com\patientis\upgrade\dbcopy\connection.postgresql.xml
C:\dev\patientis\trunk\deploy\upgrade\02data
```

#### 3.4.6 *Create Starter Database*

```
pg_dump -i -h localhost -p 5432 -U postgres -F p -O -v -f "starter_database.sql" demopos

copy starter_database.sql demo_database.sql
```

### 3.5 **Create Deployment**

#### 3.5.1 *Create files*

Run com.patientis.client.deploy.Deploy

*Move database and upgrade to version specific folder*

*Create migrate.sql*

*Update SoftwareUpgrade.pdf*

#### 3.5.2 *Upgrade qapos*

Upgrade qapos then dump

```
pg_dump -i -h localhost -p 5432 -U postgres -F p -O -v -f "demo_database.sql" qapos
```

Copy to deploy directory

#### 3.5.3 *Run Mirth*

Run interfaces.bat to start the interface engine.

Start mith-manager.exe  
Import channels

Note if the channel XML is updated remove FormIdentifierModel, FormRecordIdentifierModel,

Software	Last printed 7/15/2009 2:14:00 PM
Developer Guide	Version 0.94.001

FormRecordDetailIdentifierModel

### 3.5.4 *ReportWriter*

Add report writer to deploy code

### 3.5.5 *Setup*

Add checksetup.bat

### 3.5.6 *Files*

Add 1.zip files to the upgrade/server/data and add the 5.zip from the demo environment .

## 3.6 **Create Windows Installer**

### 3.6.1 *Run InstallJammer*

New Project: PatientOS\_<version>

Application Name: PatientOS  
Short App Name: POS  
Version: <version>  
Company: patientos.org

Install Version: <version>  
Windows app executable blank out  
UNIX app executable blank out

Application Directory C:/dev/local/deploy/yyyymddd  
Remove custom components option  
Remove setup complete options

### 3.6.2 *Configure Installer*

General→Platform→Default Destination Dir add /<% Version%>  
General→Platform→Default Program Folder add /<% Version%>  
Component & Files→Groups and Files→Add mirth manager.exe as Add to Program Folder  
Component & Files→Groups and Files→Add interfaces.bat as Add to Program Folder  
Component & Files→Groups and Files→Add appserver.bat as Add to Program Folder  
Component & Files→Groups and Files→Add patientos.bat as Add to Program Folder

Install User Interface→Install Panes and Actions→Action Groups→Install Actions→Shortcut→Advanced Properties→  
Shortcutname Start Appserver (first)  
Working directory same as target file name's directory  
  
Shortcutname Start Client (second)  
Working directory same as target file name's directory

Software	Last printed 7/15/2009 2:14:00 PM
Developer Guide	Version 0.94.001

Shortcutname            Start Interface Engine  
Working directory        same as target file name's directory

Install User Interface→Install Panes and Actions→Change the program shortcut to point to adt\_out  
Build install

#### 4. Cleanup SQL

```

delete from interface_messages;
delete from item_locations;
delete from items where item_id > 0;
delete from user_states;
delete from user_resources;
delete from patient_list_visits;
delete from med_orders where med_order_id > 0;
delete from item_locations where item_location_id > 0;
delete from items where item_id > 0;
delete from med_item_packages where med_item_package_id > 0;
delete from med_item_identifiers;
delete from identifiers where deleted_ind = 1;
delete from med_item_routes;
delete from med_items where med_item_id > 0;
delete from med_identifiers;
delete from meds;
delete from refs where reference_group = 'Medication';
delete from refs where reference_group = 'MedItem';
delete from scheduler_job_instances;
delete from scheduler_jobs;
delete from order_instances;
delete from orders where visit_id > 0;
delete from form_record_details where form_record_id in (select form_record_id from form_records where visit_id > 0);
delete from form_records where visit_id > 0;
delete from forms where visit_id > 0;
update visits set primary_insurance_policy_id = 0;
update visits set secondary_insurance_policy_id = 0;
update visits set tertiary_insurance_policy_id = 0;
delete from census where census_id > 0;
delete from forms where form_id > 0 and visit_id > 0;
delete from charges where charge_id > 0;
delete from setting_logs;
delete from resource_schedules;
delete from charge_items where charge_item_id > 0;
delete from resources where resource_id > 0;
delete from visit_logs;
delete from visits where visit_id > 0;
delete from patient_identifiers;
delete from patient_portal_users where patient_portal_user_id > 0;
delete from patients where patient_id > 0;

```

Software	Last printed 7/15/2009 2:14:00 PM
Developer Guide	Version 0.94.001

## **Create linux install**

### *4.1.1 Copy files to linux*

zip C:\dev\local\deploy\20070923\client, server and upgrade directories to “serverupgrade.zip”  
psftp <server>

login as root

```
cd /usr/local/patientos_deploy  
put serverupgrade.zip
```

```
./PACK
```

## **5. Diagnostic Tool**

Run diagnostic.bat

Run install jammer