
PatientOS

**PatientOS
Linux Installation
Technical Guide**

Version 0.99

Linux Installation	Last printed 12/19/2008 9:36:00 a12/p12
Technical Guide	Version 0.75

Revision History

Date	Version	Description	Author
01/15/2010	0.99	Software Installation of PatientOS v0.99	PatientOS Inc

Linux Installation	Last printed 12/19/2008 9:36:00 a12/p12
Technical Guide	Version 0.75

Contents

- 1. OVERVIEW.....3**
- 1.1 PROCESS.....3
- 1.2 DEFINITIONS3
- 2. SOFTWARE INSTALLATION.....4**
- 2.1 PATIENTOS DOWNLOAD.....4
- 2.1.1 *Linux*.....4
- 2.2 PATIENTOS INSTALLATION5
- 2.2.1 *Unzip*.....5
- 2.3 POSTGRESQL INSTALLATION.....5
- 2.3.1 *Software*.....5
- 3. DATABASE INSTALLATION.....6**
- 3.1 VALIDATE POSTGRESQL VERSION.....6
- 3.2 CREATE DATABASE6
- 3.3 CREATE DATABASE LOGIN.....7
- 3.4 IMPORT PATIENTOS DATABASE CONTENTS.....8
- 3.4.1 *Manual*.....8
- 4. START APPLICATION.....9**
- 4.1 APPLICATION SERVER.....9
- 4.2 CLIENT10

Linux Installation Technical Guide

1. Overview

1.1 Process

The basic steps to installing PatientOS are divided into the backend system and front end client

a) Backend System

- a. Install the PatientOS software.
- b. Install and create a PostgreSQL database version 8.2 or higher.
- c. Import the PatientOS database contents.
- d. Start the PatientOS application server.

b) Frontend Client

- a. Install the PatientOS software.
- b. Start the PatientOS client.

1.2 Definitions

POS	Refers to the PatientOS software as a system.
Java	Programming language developed by Sun
PostgreSQL	An Open Source (free) database server, an alternative Oracle, SQLServer, etc.
Application Server	The 'backend' J2EE server which clients connect to.
Client	The rich (fat) client graphic user interface (GUI) with which the users interact.

2. Software Installation

2.1 PatientOS Download

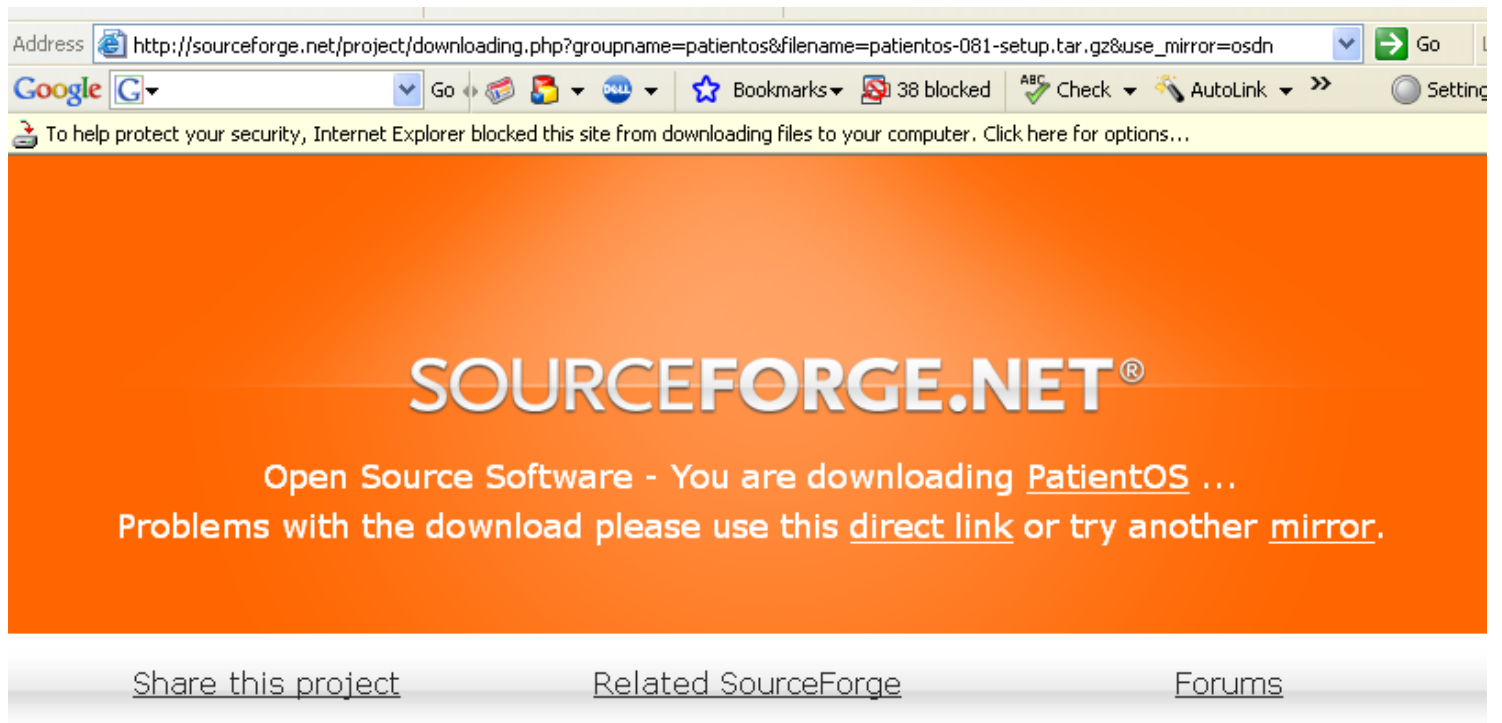
2.1.1 Linux

The PatientOS download is hosted at Sourceforge. From the command line enter the following

Linux:

```
wget http://downloads.sourceforge.net/patientos/patientos-099-setup.tar.gz
```

Links from the website redirect to sourceforge where it is often difficult to find the download link – look for the text “direct link” shown below.



2.2 PatientOS Installation

2.2.1 Unzip

Unzip the file and expand the archive in your chosen directory. A single subdirectory patientos will be created from your current directory.

```
cd /usr/local
gzip -d patientos-099-setup.tar.gz
tar -xf patientos-099-setup.tar
```

2.3 PostgreSQL Installation

2.3.1 Software

PostgreSQL is typically included in most linux distributions. If not follow your [distribution instructions](#), or execute the below command in terminal as root user.

```
yum -y install postgresql postgresql-server php-pgsql
```

If included it may not be setup to run automatically. To setup the service to auto start execute

```
chkconfig --level 345 postgresql on
```

To check the status of the service and start it if not running execute. If PostgreSQL has not been installed execute initdb

```
service postgresql status
service postgresql start
service postgresql initdb
```

If the service command does not work execute the script directly e.g.

```
/etc/init.d/postgresql status
```

Edit the postgresql.conf file and ensure that listen_addresses does not have a # comment symbol in the line, it should read

```
/var/lib/pgsql/data/postgresql.conf
listen_addresses = '*'
```

3. Database Installation

3.1 Validate PostgreSQL version

PostgreSQL must be version 8.2 or higher

```
psql --version
```

3.2 Create Database

Before the PatientOS can start properly a database is needed with a username and password the application server can connect to. There also needs to be PatientOS specific contents in the database.

Create the demopos database with the desired encoding. You **must have a UTF8 database**.

```
su - postgres  
createdb -E UTF8 demopos
```

UTF8 is required for languages with extended character sets.

3.3 Create Database Login

Create the demopos user and grant privileges for the demopos database

```
createuser -P demopos
```

Enter the password demopos twice and then enter **y to be a super user**.

```
-bash-3.2$ createuser -P demopos
Enter password for new role:
Enter it again:
Shall the new role be a superuser? (y/n) y
CREATE ROLE
-bash-3.2$ █
```

Grant privileges for the demopos database

```
psql -t template1
grant all on database demopos to demopos;
\q
```

Edit /var/lib/pgsql/data/pg_hba.conf and add lines for demopos to grant access to connect to the database on localhost

```
/var/lib/pgsql/data/pg_hba.conf

local all all password
host all all 127.0.0.1/32 password
host all all ::1/128 password
```

password is needed so that the user can login without being the same operating system user name

Any changes to pg_hba.conf and you must restart the server (logged in as root)

```
/etc/init.d/postgresql stop
/etc/init.d/postgresql start
```

Remove existing installation

If you have a previous version installed you can do the following

```
psql -U demopos demopos
drop schema public cascade;
create schema public;
\q
```

The database.sql file can then be reloaded.

Validate that the user demopos can login using psql with the host option

```
psql -h 127.0.0.1 -U demopos demopos
```

3.4 Import PatientOS Database Contents

Starter Database

3.4.1 Manual

Download the additional files

```
wget http://downloads.sourceforge.net/patientos/starter-099-database.zip
unzip starter-099-database.zip
psql -U demopos demopos < demo_database.sql

cd /usr/local/patientos/099/server/data
wget http://downloads.sourceforge.net/patientos/patientos-099-datafiles.zip
unzip patientos-099-datafiles.zip
```

4. Start Application

4.1 Application Server

The application server can now be started. The following command will run the server in the background until stopped.

Create a user `patientos` and assign ownership to the file

```
useradd patientos
mkdir /home/patientos
chown patientos /home/patientos
usermod -d /home/patientos patientos
chown -R patientos /usr/local/patientos/099
```

```
su - patientos
cd /usr/local/patientos/099/server/appserver/bin
nohup ./appserver.sh > appserver.log 2>&1 &
```

If the processor is less than 2.2ghz the startup could take a few minutes. Monitor the log file until the line

```
tail -f appserver.log
```

Wait until the following output is seen (the last line is a scheduler message – not shown)

```
:06,280 INFO [Config] Properties {com.patientis.model.security.ApplicationControlModel.capacity=51
, com.patientis.model.clinical.RecordItemModel.cron=* * 31 Feb *, com.patientis.model.scheduling.Re
ity.ApplicationControlModel.cron=* * 31 Feb *, com.patientis.model.clinical.RecordItemModel.capacit
del.refresh.period=4000, com.patientis.model.scheduling.ResourceModel.cron=* * 31 Feb *, com.patien

:06,280 INFO [GeneralCacheAdministrator] Constructed GeneralCacheAdministrator()
:06,280 INFO [GeneralCacheAdministrator] Creating new cache
:16,480 INFO [SessionFactoryObjectFactory] Not binding factory to JNDI, no JNDI name configured
:16,483 INFO [NamingHelper] JNDI InitialContext properties: {}
```

4.2 Client

You can now start the client which will connect to the application server. If you are running a terminal session from a windows operating system start your [x-windows client](#). From the command line setup your DISPLAY to direct which x-windows client will be used. Change to the client/bin directory and start patientos.

```
DISPLAY=193.168.1.100:0
export DISPLAY
cd /usr/local/patientos/099/client/bin
./patientos.sh
```

Login with the username password of either

demo / demo

or

admin / admin

The password is case sensitive.