

Using Conquest on LINUX – version 1.4.19beta3

The server core (**dgate.exe** = **dgate** under Linux) compiles and runs on Linux systems and Solaris. I develop primarily under Windows, but currently I test the code and scripts under Linux Mint 17 in a virtual machine. I also had the server compiled on a Raspberry Pi but without a lot of the extras.

The Linux release of the server core works default with SQLite driver built into the server (no ODBC). The DbaseIII driver is also supported. Piotr Filipczuk has added a PostgreSQL driver. The native MySQL interface also can be used. The graphical user interface has not been ported to Linux, but the WEB interface is provided. In this version, most options have been well tested – it is a stable release.

To use the server, one needs a valid version of the configuration files and put them in the same directory as the dgate executable. The easiest way to do this is to unpack dicomserver1419beta3.zip with “unzip dicomserver1419beta3.zip”.

INSTALLATION

Prerequisites: 1) a running Linux system. 2) sudo installed and enough rights to perform sudo. If not, the script will not be able to install the server as web service for apache and you need to copy the files by hand. 3) Installed G++; 4) Check /usr/lib/cgi-bin/ exists and is enabled in apache2.conf.

These packages needed to be installed in a plain Linux Mint17 release for a release using SQLite:

```
sudo apt-get update
sudo apt-get install g++
sudo apt-get install apache2
sudo a2enmod cgi
sudo service apache2 restart
```

To enable postgres if you choose this to use this (not needed if using SQLite):

```
sudo apt-get install libpq-dev
sudo apt-get install postgres-xc
sudo apt-get install pgadmin3
```

The following steps illustrate a minimal installation (maklinux may need adjustments for your local installation):

(ps)ftp the zip file to linux system (e.g., into your home directory)	get the files there
mkdir conquest	
cd conquest	to there
unzip ../dicomserver1419beta3.zip	unpack all files
cd src/dgate	
chmod 777 jpeg-6c/configure	
chmod 777 maklinux	
./maklinux	compile and install web access
dgate -v -r	regenerate the database

`dgate -v &`

run the server (for ever)

Now the server should be running and `localhost/cgi-bin/dgate` should provide a working web interface.

ZeroBraneStudio IDE

To install and use ZeroBrane Studio with the conquest DICOM server under Linux, take these steps. First download `ZeroBraneStudioEduPack-xxx-linux.sh`. Then in a command prompt run:

```
chmod 777 ZeroBraneStudioEduPack-xxx-linux.sh
sudo ./ZeroBraneStudioEduPack-xxx-linux.sh
```

After installation is done run ZeroBrane Studio from the command prompt as “`sudo zbstudio`” and run the install script `/dicomserver/ZeroBraneStudio/install.lua` in ZeroBrane Studio as described in this file. After running the conquest install script as root, ZeroBraneStudio can be run as a normal user.

Some of the scripts make use of external binaries (normally not needed):

I installed `cmake` and `cmake-qt-gui` to build and install `nifty_reg`
Then I installed `mricon` as 'small' nifti viewer
and finally `p7zip.full` to enable use of `7za` as decompressor

CONFIGURATION

Configuration files under Windows and Linux are the same except for the use of a forward slash instead of back slash in directory paths. The following essential entries are therefore different for Linux (these are the defaults):

SQLServer	=	./data/dbase/conquest.db3
MAGDevice0	=	./data/

See the Windows manual for more details about the configuration files (you need at least to edit **acrnama.map** to define DICOM systems that will be retrieving information from your server). All configurations options in **dicom.ini** (e.g., for DICOM routing) are listed in **windowsmanual.pdf**. You probably also need to edit the web server configuration file **/usr/lib/cgi-bin/dicom.ini** to set the correct IP address of the machine. If not the web server will only partly function.

After copying the files, if needed, regenerate the database with “conquest/dgate -v -r” then run the server with “conquest/dgate -v” or “conquest/dgate -^serverstatus.log”. NOTE: regeneration is only needed after an upgrade if **dicom.sql** is updated. If you want to avoid regeneration do NOT replace **dicom.sql**

To automatically start the server at boot time create a shell script in /etc/rc5.d called Z99Conquest, that contains, e.g.,:

```
cd /home/marcel/conquest
dgate -^serverstatus.log
```

The building process for the server was tested with gcc 3.3.5, Ubuntu 8.10 and on Solaris 10. Both 32 and 64 bit OS's are supported. Warnings (many ‘multi-character character constant’ and one ‘fattach is not implemented and will always fail’) are produced but these do not impact server operation.

Shell script **maklinux** is available that compiles dgate, copies it to the cgi-bin directory for web access, and sets up (*overwrites*) **dicom.ini** and **dicom.sql** for SQLite operation. The SQLite driver is built-in.

Also a shell script **maklinux_dbase** is available that compiles dgate with dbaseIII support and copies it to the cgi-bin directory for web access. It also sets up (*overwrites*) **dicom.ini** and **dicom.sql** for dbaseIII operation. The dbaseIII driver is built-in.

Also a shell script **maklinux_mysql** is available that compiles dgate with MySQL support and copies it to the cgi-bin directory for web access. It also sets up (*overwrites*) **dicom.ini** and **dicom.sql** for SQLite operation. It requires creating a DB called "conquest" with phpmyadmin and installing libmysqlclientdev with: “*apt-get install libmysqlclient-dev*” before running maklinux_mysql. These are the settings in dicom.ini for MySQL:

SQLHost	=	localhost
SQLServer	=	conquest
Username	=	root
Password	=	
Mysql	=	1

DoubleBackSlashToDB = 1

Also a shell script **maklinux_postgres** is available that compiles dgate and copies it to the cgi-bin directory for web access. It also makes sure the postgres shared libraries can be found, and sets up (*overwrites*) **dicom.ini** and **dicom.sql** for PostGres operation. The PostGres system (I used postgresql-8.1beta1.tar.bz2) must be setup to the defaults, and a database named '*conquest*' made. For postgres to work you need to check some values in dicom.ini (using the default postgres account assuming password postgres, note that parameter '*SQLServer*' sets the database to conquest). A copy from **dicom.ini.postgres** to **dicom.ini** would set the following values:

```
SQLHost          = localhost
SQLServer        = conquest
Username         = postgres
Password         = postgres
PostGres         = 1
DoubleBackSlashToDB = 1
UseEscapeStringConstants = 1
```

It is advised to use a normalized database (as defined in **dicom.sql**) for postgres operation, e.g., by copying **dicom.sql.postgres** to **dicom.sql** and a denormalized database for DbaseIII, e.g., by copying **dicom.sql.dbase** to **dicom.sql**. The following are donated scripts by Mark Pearson for start/stop and rotating logfiles:

To install this script (it is in the distribution as conquest-pacs.sh) do:

```
sudo cp conquest-pacs.sh /etc/init.d/
sudo chmod 755 /etc/init.d/conquest-pacs.sh
sudo apt-get install authbind
sudo /etc/init.d/conquest-pacs.sh start
```

```
#!/bin/bash
#
# conquest-pacs.sh          SysV init script for Conquest PACS.
#
#       Written by Miquel van Smoorenburg <miquels>.
#       Modified for Debian GNU/Linux by Ian Murdock <imurdock>.
#       Customized for Conquest by Mark Pearson <markp>
#
#       HOME and PACSUSER should be the only variables that may need to be
modified.
#
PATH=/sbin:/bin:/usr/sbin:/usr/bin

# Modify HOME to suit your environment.
HOME=/usr/local/conquest
# This is the user to run as. Modify it if you don't use username conquest.
PACSUSER=conquest

DAEMON=$HOME/dgate
INI=$HOME/dicom.ini
NAME=conquest_pacs.sh
```

```

# All defaults here will be overridden by values from $HOME/dicom.ini
STATUSLOG=$HOME/serverstatus.log
PORT=104
DESC="Conquest PACS Server"

STOPPACS=$HOME"/dgate --quit:"
STARTAS=$DAEMON

test -f $DAEMON || echo "Cannot find $DAEMON" exit 0
test -f $INI || echo "Cannot find $INI" exit 0

set -e

if grep "TCPPort" $INI > /dev/null ; then
    PORT=`egrep -i '^*TCPPort *= ' $INI | sed 's/\r//' | awk '{ print $3}'`
fi

if [ $PORT -le 1024 ]; then
    test -f /usr/bin/authbind || echo "authbind is needed for access to ports <
1024" exit 0
    STARTAS="/usr/bin/authbind "
fi

if grep -is "^ *StatusLog" $INI > /dev/null ; then
    STATUSLOG=`egrep -i '^*StatusLog' $INI | sed 's/\r//' | awk '{ print
$3}'`
fi

PIDFILE=/var/run/$NAME.$PORT.pid
if [ $STARTAS = $DAEMON ]; then
    ARGS=" -^$STATUSLOG"
else
    ARGS="$DAEMON -^$STATUSLOG"
fi

case "$1" in
    start)
        if [ -f $HOME/disable_autostart ]; then
            echo "Not starting $DESC: disabled via $HOME/disable_autostart"
            exit 0
        fi

        echo -n "Starting $DESC: "
        start-stop-daemon --start --quiet --pidfile $PIDFILE \
            --chuid $PACSUSER --chdir $HOME --exec $DAEMON \
            --startas $STARTAS --background -- $ARGS
        echo "$NAME."
        ;;
    stop)
        echo -n "Stopping $DESC: "
        cd $HOME
        $STOPPACS

        start-stop-daemon --oknodo --stop --quiet --pidfile $PIDFILE \
            --exec $DAEMON -- $ARGS
        echo "$NAME."
        echo
    esac

```

```

;;

restart|force-reload)
    echo -n "Restarting $DESC: "
    start-stop-daemon --stop --oknodo --quiet --pidfile $PIDFILE \
        --exec $DAEMON -- $ARGS
    sleep 1
    start-stop-daemon --start --quiet --pidfile $PIDFILE \
        --chuid conquest --chdir $HOME --exec $DAEMON -- $ARGS
    echo "$NAME."
    ;;
*)
    N=/etc/init.d/$NAME
    echo "Usage: $N {start|stop|restart|force-reload}" >&2
    exit 1
    ;;
esac

exit 0

```

For security reasons I have added a user "conquest" and the package authbind to allow access to privileged ports. I added the following entries to dicom.ini:

HomeDir = /usr/local/conquest

StatusLog = /var/log/conquest/NMPACS.serverstatus.log

TroubleLog = /var/log/conquest/NMPACS.PacsTrouble.log

The file /etc/cron.weekly/conquest_rotate does weekly log rotation for me.

```

#!/bin/bash

# conquest_rotate      Cron script to rotate conquest log files.
#      Keep files for 365 days
#      Read filenames from dicom.ini
#
#
#      Written by Mark Pearson 20070711 <markp>.
#

# Modify this line to suit your environment
HOMES=(/usr/local/conquest /usr/local/conquest-icon)
for i in ${HOMES[@]}; do

    INI=${i}/dicom.ini
    STATUSLOG=${i}/serverstatus.log
    TROUBLELOG=${i}/PacsTrouble.log

    set -e

# defaults will be overridden by values from ${i}/dicom.ini
    if grep -is "^ *StatusLog" $INI > /dev/null ; then
        STATUSLOG=`egrep -i '^*StatusLog' $INI | sed 's/\r//'` | awk
    '{ print $3}'`
    fi
    if grep -is "^ *TroubleLog" $INI > /dev/null ; then
        TROUBLELOG=`egrep -i '^*TroubleLog' $INI | sed 's/\r//'` | awk
    '{ print $3}'`

```

```

fi

if [ -s $TROUBLELOG ]; then
    savelog -p -c 365 -n -q $TROUBLELOG
fi

if [ -s $STATUSLOG ]; then
    savelog -p -c 365 -n -q $STATUSLOG
fi
done

```

This copes with multiple pacs instances on the same host. The advantage of using savelog is that old logfiles are compressed. It should be quite simple to edit the files to have executable or log in /opt. Also, don't forget to set the appropriate file permissions for the user that runs conquest.

Finally, Here are the command lines to compile the server under OS X xcode using 10.4u sdk on a PowerPC:

```

g++ -isysroot /Developer/SDKs/MacOSX10.4u.sdk -arch ppc -Wno-multichar
-I/usr/local/mysql/include -L/usr/local/mysql/lib -DDARWIN -DUSEMYSQL -DHAVE_LIBJASPER
-DHAVE_LIBJPEG -DB_DEBUG -o dgate total.cxx -lpthread -lgcc_s.10.4 -lstdc++.6 -lmysqlclient
-lz

```

And to compile under SOLARIS 10:

```

/usr/sfw/bin/g++ -DUNIX -DNATIVE_ENDIAN=1 -DHAVE_LIBJASPER -DHAVE_LIBJPEG
-DSOLARIS total.cxx -o dgate -lpthread -lsocket -lnsl -lposix4

```