

## Using Conquest on LINUX – version 1.4.14

Since server version 1.4.8, the server core (**dgate.exe** = **dgate** under Linux) compiles and runs on a Linux system. I have developed and tested the code under Knoppix, which is a Debian Linux version that boots and runs from a CD. I installed Knoppix both on hardisks and in virtual machines.

The Linux release of the server core works default with the indexed dBaseIII driver built in into the server (no ODBC) and the internal jpeg decompression engine is not included. Piotr Filipczuk has added a PostGRES SQL driver that has been reasonably tested . Since version 1.4.13, SQLite 3.4.0 has been added as database engine as well. A native mysql driver is possible (manual from beral). 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 CONQUESTLINUX1414.ZIP with “unzip conquestlinux1414.zip” that contains:

conquest/dgate	Server executable
conquest/dicom.ini	Server configuration (for dbase)
conquest/dicom.ini.dbase	Template configuration for built in dbase driver
conquest/dicom.ini.sqlite	Template configuration for built in sqlite driver
conquest/dicom.ini.postgres	Template server configuration for postgres
conquest/dicom.ini.mysql	Template server configuration for postgres
conquest/dicom.ini.www	Template configuration for web server
conquest/dicom.sql	Database config (denormalized, for dbase)
conquest/dicom.sql.dbase	Template database configuration (denormalized)
conquest/dicom.sql.postgres	Template database configuration (normalized)
conquest/dicom.sql.sqlite	Template database configuration (normalized)
conquest/dicom.sql.mysql	Template database configuration (normalized)
conquest/acrnema.map	Configuration of know DICOM providers
conquest/dgate.dic	DICOM dictionary
conquest/dgatesop.lst	Accepted data and services
conquest/dgatesop.lst.nojpg	Template accepted data and services (no jpeg)
conquest/dgatesop.lst.withjpg	Template accepted data and services (with jpeg)
conquest/maklinux	Shell script to compile and install dgate
conquest/maklinux_postgres	Idem but using Postgres as database
conquest/maklinux_sqlite	Idem but using SQLite as database
conquest/maklinux_mysql	Idem but using MySQL as database
conquest/maklinux.bat	Used by mvh for collecting files in Linux distribution.
conquest/Makefile	Sample Makefile (unused)
conquest/ActiveFormProj1.ocx	ActiveX control (IE only) for distribution by web server
conquest/sample.cq	Sample page script for web server
conquest/DicomConformance_FilesLST_Changes.pdf	Part of manual
conquest/windowsmanual.pdf	Part of manual
conquest/linuxmanual.pdf	Part of manual (this file)
conquest/data/dbase/DICOMImages.DBF	Sample database (dbaseIII)
conquest/data/dbase/DICOMPATIENTS.DBF	
conquest/data/dbase/DICOMSeries.DBF	

conquest/data/dbase/DICOMStudies.DBF  
conquest/data/HEAD\_EXP\_00097038/ Sample images  
conquest/data/HEAD\_EXP\_00097038/0001\_002000\_892665661.v2  
conquest/data/HEAD\_EXP\_00097038/0001\_003000\_892665662.v2

Plus all the sources needed to build the server: aaac.cxx, aaac.hpp, aarj.cxx, aarj.hpp, aarq.cxx, aarq.hpp, amap.cpp, array.tcc, array.thh, base.hpp, buffer.cxx, buffer.thh, cctypes.h, constant.h, dbssl.cpp, deivr.cxx, deivr.hpp, device.cpp, dgate.cpp, dgate.hpp, dgatefn.cpp, dicom.hpp, dimsec.cxx, dimsec.hpp, dimsens.cxx, dimsens.hpp, dprintf.cpp, dprintf.hpp, endian.cpd, endian.cxx, endian.hpd, endian.hpp, farray.thh, filepdu.cxx, flpdu.cxx, flpdu.hpp, gpps.cpp, gpps.hpp, lex.cpp, lex.hpp, loadddo.cpp, nkiqrsop.cxx, nkiqrsop.hpp, npipe.cpp, npipe.hpp, odbci.cpp, odbci.hpp, parse.cpp, pdata.cxx, pdata.hpp, pdu.cxx, pdu.hpp, queue.tcc, pqueue.thh, qrsop.cxx, qrsop.hpp, regen.cpp, rtc.cxx, rtc.hpp, safemem.h, socket.cxx, socket.hpp, sqlite3.c, sqlite3.h, storage.cxx, storage.hpp, total.cxx, trnsyn.cxx, uniq.cxx, unixsock.h, util.cxx, util.h, verify.cxx, verify.hpp, version.h, vrtosql.cpp, wintypes.hpp, xvgifwr.c

## 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) Install G++; 4) Check /usr/lib/cgi-bin/ exists and is enabled in apache2.conf.

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

(ps)ftp zips to linux system (e.g., into your home directory)	get the files there
unzip conquestlinux1414.zip	unzip all files
cd conquest	to there
chmod 777 maklinux	make scripts runnable
chmod 777 maklinux_sqlite	
chmod 777 maklinux_postgres	
chmod 777 maklinux_mysql	
maklinux	compile and install web access
(or: maklinux_sqlite, or: maklinux_postgres, or: maklinux_mysql)	
(optional for JPEG support)	if JPEG support needed
cd ..	
unzip jpegsuplinux1414.ZIP -d conquest	
chmod 777 conquest/dcmdjpeg	
chmod 777 conquest/dcmcjpeg	
cd conquest	
(end optional)	
dgate -v -r	regenerate the database
dgate -v &	run the server (for ever)

## 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/
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 **/usr/lib/cgi-bin/dicom.ini** to set the correct IP address of the machine. If not the web server will not fully 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 and on Solaris 10. 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. NOTE 1: if you are running on a 64 bits linux version you should add **-m32** to the g++ command line as only 32 bits is supported for now. Note 2: The conquest server code has been optimized on windows. It runs almost a factor of 2 slower on linux with the same hardware.

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 DbaseIII operation.

Also a shell script **maklinux\_sqlite** is available that compiles dgate with SQLite 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. The SQLite 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

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:

---

```
#!/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
        ;;
    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           fi

---

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 is 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 -DNOINTJPEG  
-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 -DNOINTJPEG -DSOLARIS total.cxx -o dgate  
-lpthread -lsocket -lnsl -lposix4
```