Eloquence Workshop - Student's Guide

Revision 2010-04-16.a

Introduction

This collection of labs will provide you with some hands-on experience with various aspects of setting up, using and maintaining the Eloquence database system. It is divided into three modules:

The first one focuses on getting started with setup and usage; you will first configure and start an Eloquence database server instance, then export a small TurboImage sample database on MPE/iX, and finally import this data into Eloquence and access it with Query3k. Optionally, you can also bring a small C or COBOL sample program from MPE/iX to the Eloquence platform to see how this is compiled, linked and run with the appropriate libraries to work with Eloquence.

The MPE/iX parts of the labs are optional, in case you do not have access to an HP 3000 system.

The second module deals with various administration tasks for your Eloquence server; you will practice different methods of backup and recovery (offline, online and forward logging), examine tools for monitoring and troubleshooting server and client, perform structural and security changes to your database, and optionally look at facilities for database auditing.

The third module discusses advanced topics like performance metrics and database replication.

Most of the screen snippets in this document are taken from the Linux version of Eloquence. However, they do apply to HP-UX and Windows with only minor differences. Those differences will be mentioned where appropriate.

Table of Contents

Module 1 – Setup & Usage

Lab 1.1 – Create and start an Eloquence instance for your team

As we plan to have multiple (teams of) students working on the labs on a shared system, we are using the Eloquence feature of running multiple independent database server instances. Each team will use their own Unix logon and create config and database volume files in or below their home directory. They will run their server under their own logon, using a team-specific tcp port number.

The main Eloquence config file is prepared by the teacher. It references the individual team server instances and assigns the respective config filename and tcp port number (or better: service name).

Check out the main Eloquence server config file for references to your team's instance. Copy the default eloqdb.cfg file to the appropriate target name and customize it to your needs. You should change at least the UID and GID parameters, but also consider providing a custom logfile location and logging level. If the main Eloquence config does not override the service name (tcp port), you need to specify one in your config file. If there is an entry in /etc/services, prefer using the service name instead of the tcp port number.

After creating your custom config file, allocate at least one data and log volume with dbvolcreate and dbvolextend. Notice that the path names will be added to the end of your config file. Start the database server using the init.d script and verify its status. Your server processes should also show up in the Unix ps command.

As a first test for accessing your server via its tcp service, you can use the dbdumpcat utility (more on that later). While dbdumpcat has a command line option for the service name / port number, you should better export an appropriate EQ_DBSERVER environment variable for the remaining labs.

Note that the screen snippets below refer to team1. Adjust for your teamN as appropriate.

```
team1@unix:~> more /etc/sysconfig/eloquence8    # /etc/rc.config.d/... for hp-ux

 (...)
ELOQDB_CFG[0]=/etc/opt/eloquence/8.0/eloqdb.cfg
ELOQDB_START[0]=1
ELOQDB_ARGS[0]=""
ELOQDB_SERVICE[0]=""
ELOQDB_ID[0]="teacher"
ELOQDB_RUNPFX[0]=""

ELOQDB_CFG[1]=/home/team1/eloqdb.cfg
ELOQDB_START[1]=0
ELOQDB_ARGS[1]=""
ELOQDB_SERVICE[1]="eqdb_team1"
ELOQDB_ID[1]="team1"
ELOQDB_RUNPFX[1]=""
 (...)


team1@unix:~> more /etc/services

 (...)
eqdb_team1  3001/tcp  # team1 db server
eqdb_team2  3002/tcp  # team2 db server
eqdb_team3  3003/tcp  # team3 db server
eqdb_team4  3004/tcp  # team4 db server
 (...)
```

```
team1@unix:~> pwd

/home/team1

team1@unix:~> cp /opt/eloquence/8.0/newconfig/config/eloqdb.cfg .

team1@unix:~> chmod u+w eloqdb.cfg ; ll

-rw-r--r--  1 team1 users 21465 2005-11-24 14:42 eloqdb.cfg


team1@unix:~> ## customize eloqdb.cfg (see diff output below for changes)


team1@unix:~> diff /opt/eloquence/8.0/newconfig/config/eloqdb.cfg eloqdb.cfg

41c41
< #Service = eloqdb
---
> Service = eqdb_team1
66,67c66,67
< UID = eloqdb
< GID = eloqdb
---
> UID = team1
> GID = users
114c114
< #LogFile = syslog
---
> LogFile = /home/team1/eloqdb.log
172c172
< #LogFlags = *0
---
> LogFlags = *0E1


team1@unix:~> mkdir db; cd db


team1@unix:~/db> type dbvolcreate

dbvolcreate is /opt/eloquence/8.0/bin/dbvolcreate


team1@unix:~/db> dbvolcreate -help

ELOQUENCE DBVOLCREATE (C) Copyright 2008 Marxmeier Software AG (B.08.00)
usage: dbvolcreate [options] volume_file_name
options:
 -v        - verbose
 -d flags  - debug flags
 -c cfg    - configuration file name
 -s sz     - Initial size
 -e sz     - Extension size
 -m sz     - Max volume size


team1@unix:~/db> dbvolcreate -v -c $HOME/eloqdb.cfg data-01.vol

ELOQUENCE DBVOLCREATE (C) Copyright 2008 Marxmeier Software AG (B.08.00)
Reading configuration ...
Creating volume /home/team1/db/data-01.vol
Initializing volume ...
Closing volume ...
done.
```

```
team1@unix:~/db> dbvolextend -help

ELOQUENCE DBVOLEXTEND (C) Copyright 2008 Marxmeier Software AG (B.08.00)

usage: dbvolextend [options] volume_file_name
       dbvolextend [options] -R
options:
 -R        - recreate missing LOG volumes if possible
 -t type   - Volume type (DATA, LOG)
 -v        - verbose
 -d flags  - debug flags
 -c cfg    - configuration file name
 -s sz     - Initial size
 -e sz     - Extension size
 -m sz     - Max volume size


team1@unix:~/db> dbvolextend -v -t log -c $HOME/eloqdb.cfg log-01.vol

ELOQUENCE DBVOLEXTEND (C) Copyright 2008 Marxmeier Software AG (B.08.00)

Reading configuration ...
Opening root volume
Creating extension volume /home/team1/db/log-01.vol
Closing volume ...
done.


team1@unix:~/db> ls -l

-rw-------  1 team1 users 2621440 2005-11-24 14:56 data-01.vol
-rw-------  1 team1 users 2621440 2005-11-24 14:56 log-01.vol


team1@unix:~/db> cd

team1@unix:~> tail -5 eloqdb.cfg

### Data base environment

[Volumes]
Root = /home/team1/db/data-01.vol
Log02 = /home/team1/db/log-01.vol


team1@unix:~> /etc/init.d/eloq8 start team1    # /sbin/init.d/... for hp-ux

Starting eloqdb[team1] daemon                                     done


team1@unix:~> ps -f -u team1

UID        PID  PPID  C STIME TTY          TIME CMD
team1     8418  8417  0 14:14 pts/1    00:00:00 /bin/bash
team1     8849     1  0 15:00 pts/1    00:00:00 eloqdb32: active
team1     8884  8418  0 15:01 pts/1    00:00:00 ps -f -u team1


team1@unix:~> /etc/init.d/eloq8 status    # /sbin/init.d/... for hp-ux

eloqsd process is not active                                      unused
eloqdb[teacher] process is active (pid 4575)                      running
eloqdb[team1] process is active (pid 9176)                        running
```

```
team1@unix:~> export EQ_DBSERVER=localhost:eqdb_team1

team1@unix:~> dbdumpcat -l

ELOQUENCE DBDUMPCAT (C) Copyright 2008 Marxmeier Software AG (B.08.00)
------------------------------------------------
| Server catalog                               |
| localhost:eqdb_team1                         |
|----------------------------------------------|
| Id   | Name                           | Count |
|----------------------------------------------|
|   20 | sysobjects                     |    11 |
|   21 | sysdevices                     |     2 |
|   22 | sysvat                         |     0 |
|   30 | sysuser                        |     2 |
|   31 | sysdb                          |     0 |
|   32 | systables                      |    10 |
|   33 | syscolumns                     |    65 |
|   34 | sysindex                       |     0 |
|   35 | sysindexseg                    |     0 |
|   36 | syscollate                     |     0 |
------------------------------------------------
```

Lab 1.2 – Migrate small TurboImage database from MPE to Eloquence

In this exercise we will migrate a small sample database from MPE/iX to Eloquence using the dbexport and dbimport utilities. You will export the data on MPE, transfer database schema and unload files to the Eloquence system, create and populate an equivalent database in your team's server instance, and finally examine the results using the query3k utility.

Note: If you do not have access to an MPE system, the teacher can provide the export files to you.

Step (a): Examine the MPE sample database and programs

To familiarize yourself with the starting point, examine the TOYDB database files. The schema file is called TOYDBS. Open the database with QUERY to view FORM SETS. For the lab we will use the PRODUCTS dataset, so you should also look at FORM PRODUCTS and LIST PRODUCTS.

There is a small sample program that performs a serial read against the PRODUCTS dataset. Its source code is provided in C and COBOL, files IMG01CXL and IMG01COB, respectively. The file IMG01PRG is a pre-compiled version that you can run, even if you don't have a compiler at hand.

Use QUERY to add a team-specific entry to the PRODUCTS dataset before exporting the data.

```
:listf toydb@,2

ACCOUNT=  ELOQ        GROUP=  TEAM1

FILENAME  CODE  ------------LOGICAL RECORD-----------  ----SPACE----
                  SIZE  TYP        EOF      LIMIT R/B  SECTORS #X MX

TOYDB     PRIV    128W  FB          10         10   1       16  1  1
TOYDB01   PRIV    640W  FB           5          5   1       32  1  1
TOYDB02   PRIV    640W  FB          13         13   1       80  1  1
TOYDB03   PRIV    384W  FB          36         36   1      112  1  1
TOYDB04   PRIV    512W  FB          28         28   1      128  1  1
TOYDB05   PRIV    512W  FB          17         17   1       80  1  1
TOYDB06   PRIV    640W  FB          29         29   1      160  1  1
TOYDBS             80B  FA          83         83   3       32  1  1


:print toydbs
BEGIN DATA BASE TOYDB ;

PASSWORDS:
  10  reading ;
  20  writing ;

ITEMS:
  ADDRESS          , X30(10/20);
  AMOUNT           , P12(10/20);
  CITY             , X30(10/20);
  CUSTOMER-NAME    , X30(10/20);

 (...)


:query
HP32216D.03.21  QUERY/NM  THU, NOV 24, 2005, 11:41 AM
COPYRIGHT HEWLETT-PACKARD CO. 1976

>base=toydb
PASSWORD = >>
MODE = >>5
```

```
>form sets

DATA BASE: TOYDB                        THU, NOV 24, 2005, 11:41 AM

DATA BASE LANGUAGE ATTRIBUTE: NATIVE-3000

                            ITEM   CURRENT   ENTRY    ENTRY   BLOCKING
SETS:                 TYPE  COUNT  CAPACITY  COUNT    LENGTH  FACTOR

   PRODUCTS           M     5      100       10       15      24
   CUSTOMERS          M     10     50        8        130     4
   ORDER-MASTER       A     1      503       15       4       14
   INVOICES           D     7      504       1        20      18
   ORDERS             D     3      510       15       9       30
   ORDER-DETAILS      D     4      1015      15       10      35

>form products

DATA BASE: TOYDB                        THU, NOV 24, 2005, 11:42 AM

DATA BASE LANGUAGE ATTRIBUTE: NATIVE-3000

SET NAME:
   PRODUCTS,MANUAL

      ITEMS:
         PRODUCT-NO,          X6            <<KEY ITEM>>
         PRODUCT-NAME,        X16
         PRICE,               P8
         PRODUCT-LINE,        X2
         QUANTITY,            I1

      CAPACITY: 100            ENTRIES: 10

>list products

PRODUC  PRODUCT-NAME         PRICE  PR  QUANTI

A00003  POKER DICE SET         125  10     500
A00008  POSTER PAINTS           95  30    1250
A00009  COLOURING BOOK          65  30    1000
A00001  PACK OF CARDS           75  10    1500
A00010  ERASER GIFT SET        185  30    1500
A00002  LUDO SET              1250  10     750
A00005  15" PINK RABBIT       1745  20     200
A00007  SET OF CRAYONS         175  30     500
A00004  12" TEDDY BEAR        1525  20     250
A00006  SET OF PANDAS         2500  20     150

>exit


:listf img01@,2

ACCOUNT=  ELOQ        GROUP=  TEAM1

FILENAME  CODE  ------------LOGICAL RECORD-----------  ----SPACE----
                  SIZE  TYP        EOF      LIMIT R/B  SECTORS #X MX

IMG01COB           80B  FA         108        108   3       48  1  1
IMG01CXL           80B  FA         118        118   3       48  1  1
IMG01PRG  NMPRG   128W  FB         131        131   1      144  1  *


:# img01cob source can be re-compiled with :cob85xlk img01cob,img01prg
:# img01cxl source can be re-compiled with :ccxllk   img01cxl,img01prg
```

```
:run img01prg

opening db...
listing products...
A00003  POKER DICE SET    +0500
A00008  POSTER PAINTS     +1250
A00009  COLOURING BOOK    +1000
A00001  PACK OF CARDS     +1500
A00010  ERASER GIFT SET   +1500
A00002  LUDO SET          +0750
A00005  15" PINK RABBIT   +0200
A00007  SET OF CRAYONS    +0500
A00004  12" TEDDY BEAR    +0250
A00006  SET OF PANDAS     +0150
closing db...

END OF PROGRAM


:query
HP32216D.03.21  QUERY/NM  THU, NOV 24, 2005, 12:01 PM
COPYRIGHT HEWLETT-PACKARD CO. 1976

>base=toydb
PASSWORD = >>
MODE = >>1

>add products
PRODUCT-NO      =>>Team1A
PRODUCT-NAME    =>>Fun Product
PRICE           =>>399
PRODUCT-LINE    =>>X
QUANTITY        =>>1

PRODUCT-NO      =>>//

>exit
```

Step (b): Export database contents and transfer files to Unix

For our small sample database we can use dbexport without any special options, resulting in serial unload of manual master and detail sets to individual output files. For real-life databases, you might need to use „dbexport -B“ for binary export, which needs PM capability but is typically faster, takes care of the 2 GB output file limit, and is able to preserve the chronological order of all chains.

The resulting export files are in bytestream format and should be transferred to the Eloquence system using binary mode. We are using ftp here, but the file transfer feature of your favourite terminal emulator should also work. It is typically useful to also convert the schema file to bytestream format (stripping trailing blanks and inserting LF delimiters) and then transfer it in binary mode as well.

```
:listf db@.pub ,2

ACCOUNT=  ELOQ       GROUP=  PUB

FILENAME  CODE  ------------LOGICAL RECORD-----------  ----SPACE----
                  SIZE  TYP        EOF      LIMIT R/B  SECTORS #X MX

DBEXPORT  NMPRG   128W  FB         672        672   1      672  3  *
DBINFO    NMPRG   128W  FB         513        513   1      528  3  *


:dbinfo -help
usage: DBINFO [options] database [set ...]
options:
 -help    - show usage (this list)
 -version - show version information
 -u user  - set user name
 -p pswd  - set password


:dbinfo toydb
Processing database : toydb

SET NAME              RECLEN CAPACITY ENTRIES
---------------- -- - ------ -------- --------
PRODUCTS         01 M     15      100       11
CUSTOMERS        02 M    130       50        8
ORDER-MASTER     03 A      4      503       15
INVOICES         04 D     20      504        1
ORDERS           05 D      9      510       15
ORDER-DETAILS    06 D     10     1015       15


:dbexport -help
usage: DBEXPORT [options] database [set[.item] ...]
options:
 -help    - show usage (this list)
 -version - show version information
 -u user  - set user name
 -p pswd  - set password
 -o path  - set output directory (not single file)
 -v       - verbose output
 -c       - chained export
 -a       - export automatic sets
 -r       - create restructure information
 -s file  - output into single file, '-' = stdout
 -f sep   - field separator, default is ','
 -x       - exclude specified data sets
 -B       - binary export mode (needs PM)
 -S       - chained export by sort item
```

```
:dbexport "-p ; -v toydb"
Processing database : toydb
Export path         : .

DATA SET               RECORDS  COUNT
---------------- --- - -------- --------
PRODUCTS         001 M       11       11
CUSTOMERS        002 M        8        8
ORDER-MASTER     003 A       15
INVOICES         004 D        1        1
ORDERS           005 D       15       15
ORDER-DETAILS    006 D       15       15


:listfile ./@.exp ,2

 PATH= /ELOQ/TEAM1/

 CODE  ------------LOGICAL RECORD-----------  ----SPACE----  FILENAME
         SIZE  TYP        EOF      LIMIT R/B  SECTORS #X MX

           1B  BA         423 2147483647   1       16  1  *  TOYDB.001.exp
           1B  BA         983 2147483647   1       16  1  *  TOYDB.002.exp
           1B  BA          50 2147483647   1       16  1  *  TOYDB.004.exp
           1B  BA         390 2147483647   1       16  1  *  TOYDB.005.exp
           1B  BA         428 2147483647   1       16  1  *  TOYDB.006.exp


:print ./TOYDB.001.exp
"A00003","POKER DICE SET",125,"10",500
"Team1A","Fun Product",399,"X",1
"A00008","POSTER PAINTS",95,"30",1250
"A00009","COLOURING BOOK",65,"30",1000
"A00001","PACK OF CARDS",75,"10",1500

 (...)


:# convert schema to bytestream format, stripping trailing blanks

:/bin/tobyte "-at TOYDBS TOYDB.sch"



:# now transfer all files to unix using binary mode (for bytestream files)

:ftp unix
File Transfer Protocol [A0012C05] (C) Hewlett-Packard Co. 2002 [PASSIVE SUPPORT]
220 unix.demo.com FTP server (Revision 1.003 Version wuftpd-2.6.1) ready.
Connected to unix (192.x.xx.xx).  (FTPINFO 40)

Name(team1): team1
331 Password required for team1.
Password:
230 User team1 logged in.
Remote system type is UNIX

ftp> pwd
257 "/home/team1" is current directory.

ftp> mkdir tmp
257 MKD command successful.

ftp> cd tmp
250 CWD command successful.

ftp> bin
200 Type set to I.

ftp> put ./TOYDB.sch
200 PORT command successful.
150 Opening BINARY mode data connection for ./TOYDB.sch.
```

```
226 Transfer complete.
3130 bytes sent in 0.03 seconds (113.21 Kbytes/sec)

ftp> prompt
Interactive mode off.  (FTPINFO 42)

ftp> mput ./TOYDB.@.exp
200 PORT command successful.
150 Opening BINARY mode data connection for ./TOYDB.001.exp.
226 Transfer complete.
423 bytes sent in 0.02 seconds (17.21 Kbytes/sec)
200 PORT command successful.
150 Opening BINARY mode data connection for ./TOYDB.002.exp.
226 Transfer complete.
983 bytes sent in 0.02 seconds (43.63 Kbytes/sec)
200 PORT command successful.
150 Opening BINARY mode data connection for ./TOYDB.004.exp.
226 Transfer complete.
50 bytes sent in 0.02 seconds (2.03 Kbytes/sec)
200 PORT command successful.
150 Opening BINARY mode data connection for ./TOYDB.005.exp.
226 Transfer complete.
390 bytes sent in 0.02 seconds (20.05 Kbytes/sec)
200 PORT command successful.
150 Opening BINARY mode data connection for ./TOYDB.006.exp.
226 Transfer complete.
428 bytes sent in 0.02 seconds (17.42 Kbytes/sec)

ftp> bye
221-You have transferred 5404 bytes in 6 files.
221-Total traffic for this session was 6894 bytes in 7 transfers.
221-Thank you for using the FTP service on unix.demo.com.
221 Goodbye.
:
```

Step (c): Create database in your Eloquence instance and import the MPE data

Use schema -T to process the TurboImage schema definition and dbcreate to allocate the database. You can use the „dbdumpcat -t 31“ utility to view database names inside your Eloquence server instance. Use dbinfo to see details about our specific database. After running dbcreate you should see the table names, but with entry counts still being zero.

Use dbimport to populate your database with the data that has been exported on the MPE system. After running dbimport you should see the proper entry counts in the dbinfo output. (Note: If you had used „dbexport -B“ for binary export on MPE, you would have to use the „bimport“ feature of dbctl for loading the data into your Eloquence database.)

```
team1@unix:~> cd tmp; ls -l TOYDB*

-rw-r--r--  1 team1 users  423 2005-11-24 21:15 TOYDB.001.exp
-rw-r--r--  1 team1 users  983 2005-11-24 21:15 TOYDB.002.exp
-rw-r--r--  1 team1 users   50 2005-11-24 21:15 TOYDB.004.exp
-rw-r--r--  1 team1 users  390 2005-11-24 21:15 TOYDB.005.exp
-rw-r--r--  1 team1 users  428 2005-11-24 21:15 TOYDB.006.exp
-rw-r--r--  1 team1 users 3130 2005-11-24 21:17 TOYDB.sch


team1@unix:~/tmp> schema -help

ELOQUENCE SCHEMA (C) Copyright 2008 Marxmeier Software AG (B.08.00)

usage: schema [options] file
options:
 -help      - show usage (this list)
 -u user    - set user name
 -p pswd    - set password
 -h host    - host to contact
 -s service - service name or or port number
 -d flgs    - debug flags
 -b name    - specify database name
 -l         - output source listing           (LIST)
 -n         - no root file, only check syntax (NOROOT)
 -t         - output set table                (TABLE)
 -e cnt     - Abort after cnt error messages  (ERRORS=)
 -T         - HP3000 TurboImage compatibility mode
 -W width   - Limit line length to width
 -L         - Add line number to source listing
 -w id[,id] - Suppress specified warning messages


team1@unix:~/tmp> schema -T TOYDB.sch

ELOQUENCE SCHEMA (C) Copyright 2008 Marxmeier Software AG (B.08.00)
PAGE  1    TOYDB                ELOQUENCE SCHEMA PROCESSOR B.08.00

Item name count         : 23
Index item name count   : 0
Data set count          : 6
Collating sequence count: 0

*** Database catalog created
```

```
team1@unix:~/tmp> dbdumpcat -t 31

ELOQUENCE DBDUMPCAT (C) Copyright 2008 Marxmeier Software AG (B.08.00)

-------------------------------------------
#31 sysdb (1 entries)
-------------------------------------------
|dbid|name                |flags   |nodeid|
-------------------------------------------
|11  |TOYDB               |04000000|100   |
-------------------------------------------


team1@unix:~/tmp> dbcreate -help

ELOQUENCE DBCREATE (C) Copyright 2008 Marxmeier Software AG (B.08.00)

usage: dbcreate [options] database [set ...]
options:
 -help   - show usage (this list)
 -u user - set user name
 -p pswd - set password
 -d flgs - debug flags


team1@unix:~/tmp> dbcreate toydb

ELOQUENCE DBCREATE (C) Copyright 2008 Marxmeier Software AG (B.08.00)


team1@unix:~/tmp> dbinfo toydb

ELOQUENCE DBINFO (C) Copyright 2008 Marxmeier Software AG (B.08.00)

Processing database: toydb

SET NAME               RECLEN CAPACITY ENTRIES
---------------- --- - ------ -------- --------
PRODUCTS         001 M     30        0        0
CUSTOMERS        002 M    260        0        0
ORDER-MASTER     003 A      8        0        0
INVOICES         004 D     40        0        0
ORDERS           005 D     18        0        0
ORDER-DETAILS    006 D     20        0        0


team1@unix:~/tmp> dbimport -help

ELOQUENCE DBIMPORT (C) Copyright 2008 Marxmeier Software AG (B.08.00)

usage: dbimport [options] database [set ...]
options:
 -help   - show usage (this list)
 -u user - set user name
 -p pswd - set password
 -c cnt  - records per transaction
 -i path - set import path
 -v      - verbose output
 -t      - trace item value assignment
 -r file - restructure database, '-' = no file
 -s file - import from single file, '-' = stdin
 -e file - log errors instead of aborting
 -f sep  - field separator, default is ','
 -z cset - set import code set (roman8, iso88591)
 -d flgs - debug flags
 -x      - exclude specified data sets
 -T      - TurboIMAGE compatibility
```

```
team1@unix:~/tmp> dbimport -v toydb

ELOQUENCE DBIMPORT (C) Copyright 2008 Marxmeier Software AG (B.08.00)

Processing database : toydb
Import path         : .
Records/Transaction : 100

DATA SET                COUNT
---------------- --- - --------
PRODUCTS         001 M       11
CUSTOMERS        002 M        8
ORDER-MASTER     003 A
INVOICES         004 D        1
ORDERS           005 D       15
ORDER-DETAILS    006 D       15


team1@unix:~/tmp> dbinfo toydb

ELOQUENCE DBINFO (C) Copyright 2008 Marxmeier Software AG (B.08.00)

Processing database: toydb

SET NAME                RECLEN CAPACITY ENTRIES
---------------- --- - ------ -------- --------
PRODUCTS         001 M     30     1456       11
CUSTOMERS        002 M    260      229        8
ORDER-MASTER     003 A      8     1456       15
INVOICES         004 D     40     1149        1
ORDERS           005 D     18     1771       15
ORDER-DETAILS    006 D     20     1771       15


team1@unix:~/tmp> cd
```

Step (d): Examine results with Query3k for Eloquence

Use the query3k utility provided by Eloquence to examine the database that has been imported. Notice the team-specific entry in the PRODUCTS dataset that you added during step (a) of this exercise. By changing the EQ_DBSERVER environment variable, you can also examine the databases of your neighbour teams (but please be polite and don't attempt to modify their data); remember to set EQ_DBSERVER back to your own Eloquence instance for the next labs!

```
team1@unix:~> query3k

B.08.00.01 Eloquence QUERY3K  THU, NOV 24, 2005,  9:51 PM
Copyright 2004-2008 Marxmeier Software AG
Copyright 2004 Hewlett-Packard Development Company, L.P.

>base=toydb
PASSWORD = >>
MODE = >>5

>form sets

DATA BASE: TOYDB                        THU, NOV 24, 2005,  9:52 PM

DATA BASE LANGUAGE ATTRIBUTE: NATIVE-3000

                            ITEM   CURRENT   ENTRY    ENTRY   BLOCKING
SETS:                 TYPE  COUNT  CAPACITY  COUNT    LENGTH  FACTOR

   PRODUCTS           M     5      1456      11       15      0
   CUSTOMERS          M     10     229       8        130     0
   ORDER-MASTER       A     1      1456      15       4       0
   INVOICES           D     7      1149      1        20      0
   ORDERS             D     3      1771      15       9       0
   ORDER-DETAILS      D     4      1771      15       10      0

>list products

PRODUC  PRODUCT-NAME         PRICE  PR  QUANTI

A00003  POKER DICE SET         125  10     500
Team1A  Fun Product            399  X        1
A00008  POSTER PAINTS           95  30    1250
A00009  COLOURING BOOK          65  30    1000
A00001  PACK OF CARDS           75  10    1500
A00010  ERASER GIFT SET        185  30    1500
A00002  LUDO SET              1250  10     750
A00005  15" PINK RABBIT       1745  20     200
A00007  SET OF CRAYONS         175  30     500
A00004  12" TEDDY BEAR        1525  20     250
A00006  SET OF PANDAS         2500  20     150

>mode=1

>add products

PRODUCT-NO      =>>Team1B
PRODUCT-NAME    =>>Now on Unix
PRICE           =>>199
PRODUCT-LINE    =>>U
QUANTITY        =>>2

PRODUCT-NO      =>>//


>exit
```

Lab 1.3 – Migrate small C or COBOL program from MPE to Eloquence

In this (optional) exercise, you transfer the source code of a small C or COBOL sample program (that performs a serial read through one of the datasets) from MPE/iX to the Eloquence system, make a few source code adjustments to compile and link on the new platform, and finally run it against your Eloquence database.

Note: If you do not have access to an MPE system, the teacher can provide the source files to you.

The small sample programs do not call any MPE/iX intrinsics beyond the TurboImage ones. This makes it feasible to manually adjust the source code for compiling and linking on the new platform, using the Eloquence libraries to provide the database functionality.

Below you can find screen snippets for an example using GNU gcc on Linux, followed by brief notes and screen snippets for using ANSI C or MicroFocus COBOL or AcuCOBOL on HP-UX...

For GNU gcc on Linux or ANSI C on HP-UX, you mainly need to „disable“ MPE specific #pragma declarations and include the „image3k.h“ header file. When compiling and linking, you must supply the proper options to reference the Eloquence include and library directories and library names.

Detailed example using GNU gcc on Linux:

```
team1@unix:~> cd tmp

team1@unix:~/tmp> ftp mpeix.demo.com

Connected to mpeix.demo.com.
220 HP ARPA FTP Server [A0012C05] (C) Hewlett-Packard Co. 2000 [PASV SUPPORT]

Name (mpeix.demo.com:team1): team1.eloq
331 Password required for TEAM1.ELOQ.  Syntax: userpass,acctpass

Password:
230 User logged on
Remote system type is MPE/iX.

ftp> asc
200 Type set to A.

ftp> mget img@

mget img01cob [anpqy?]? y
227 Entering Passive Mode (192,x,xx,xxx,175,87)
150 File: img01cob opened; data connection will be opened
226 Transfer complete.
8856 bytes received in 00:00 (13.41 KB/s)

mget img01cxl [anpqy?]? y
227 Entering Passive Mode (192,x,xx,xxx,175,88)
150 File: img01cxl opened; data connection will be opened
226 Transfer complete.
9676 bytes received in 00:00 (16.45 KB/s)

ftp> bye
221 Server is closing command connection


team1@unix:~/tmp> ls -l img*

-rw-r--r--  1 team1 users 8748 2005-11-24 21:58 img01cob
-rw-r--r--  1 team1 users 9558 2005-11-24 21:58 img01cxl
```

```
team1@unix:~/tmp> # remove trailing blanks for convenience

team1@unix:~/tmp> sed 's/[ ][ ]*$//' img01cob > img01.cob
team1@unix:~/tmp> sed 's/[ ][ ]*$//' img01cxl > img01.c


team1@unix:~/tmp> cp img01.c img01_gnu.c

team1@unix:~/tmp> vi img01_gnu.c

 (...)


team1@unix:~/tmp> diff -c img01.c img01_gnu.c

*** img01.c     2005-12-30 14:21:55.000000000 +0100
--- img01_gnu.c 2005-12-30 14:40:20.000000000 +0100
***************
*** 1,6 ****
--- 1,7 ----

  /* image access from C -- lars appel 11.feb.02 */

+ #ifdef _MPEXL_SOURCE
  #pragma list off
  #include <stdio.h>
  #include <string.h>
***************
*** 23,28 ****
--- 24,35 ----
  #pragma intrinsic DBUNLOCK
  #pragma intrinsic DBBEGIN
  #pragma intrinsic DBEND
+ #else
+ #include <stdio.h>
+ #include <stdlib.h>
+ #include <string.h>
+ #include <image3k.h>
+ #endif

  #define End_Of_Chain 15       /* For DBGET Mode 5 */
  #define End_Of_Data_Set 11    /* For DBGET Mode 2 */


team1@unix:~/tmp> incs=/opt/eloquence/8.0/include
team1@unix:~/tmp> libs=/opt/eloquence/8.0/lib

team1@unix:~/tmp> gcc -o img01 -I$incs img01_gnu.c -L$libs -limage3k -Wl,-rpath,$libs


team1@unix:~/tmp> ./img01

opening db...
listing products...
   A00003    POKER DICE SET      500
   Team1A    Fun Product           1
   A00008    POSTER PAINTS      1250
   A00009    COLOURING BOOK     1000
   A00001    PACK OF CARDS      1500
   A00010    ERASER GIFT SET    1500
   A00002    LUDO SET            750
   A00005    15" PINK RABBIT     200
   A00007    SET OF CRAYONS      500
   A00004    12" TEDDY BEAR      250
   A00006    SET OF PANDAS       150
   Team1B    Now on Unix           2
closing db...


team1@unix:~/tmp> cd
```

Differences for using ANSI C on HP-UX (32bit, Itanium):

```
$ incs=/opt/eloquence/8.0/include
$ libs=/opt/eloquence/8.0/lib/hpux32

$ cc -o img01 -I$incs img01_ux.c -L$libs -limage3k

$ # note: linker option -rpath not needed here
```

For MicroFocus COBOL, you mainly need to remove the „intrinsic“ keyword from the CALL statements because these are MPE specific extensions. When compiling and linking, you must supply the proper options to reference the Eloquence library directories and library names...

Differences for using MicroFocus COBOL on HP-UX (32bit, Itanium):

```
$ cp img01.cob img01_mf.cob

$ vi img01_mf.cob

 (...)


$ diff img01.cob img01_mf.cob

59c59
<            CALL intrinsic "DBOPEN" USING
---
>            CALL "DBOPEN" USING
63c63
<              call intrinsic "DBEXPLAIN" using status1
---
>              call "DBEXPLAIN" using status1
76c76
<              CALL intrinsic "DBGET" USING
---
>              CALL "DBGET" USING
87c87
<              call intrinsic "DBEXPLAIN" using status1
---
>              call "DBEXPLAIN" using status1
95c95
<            CALL intrinsic "DBCLOSE" USING
---
>            CALL "DBCLOSE" USING
99c99
<              call intrinsic "DBEXPLAIN" using status1
---
>              call "DBEXPLAIN" using status1


$ libs=/opt/eloquence/8.0/lib/hpux32

$ cob -x -o img01 img01_mf.cob -L$libs -limage3k

$ ./img01
```

For AcuCOBOL, you can use a compiler option to accept MPE specific syntax features like the CALL INTRINSIC phrase. However, you need to make sure that the AcuCOBOL runtime knows how to find the external routines in the Eloquence libraries. This can either be done by relinking the runcbl binary with a customized direct.c version (see AcuCorp web site for details) or by loading the Eloquence shared library at runtime with an explicit CALL statement. The latter is done in this example...

Differences for using AcuCOBOL on HP-UX (32bit, Itanium):

```
$ cp img01.cob img01_acu.cob

$ vi img01_acu.cob

 (...)


$ diff -c img01.cob img01_acu.cob

*** img01.cob           Fri Dec 30 06:12:51 2005
--- img01_acu.cob       Fri Dec 30 06:26:47 2005
***************
*** 6,11 ****
--- 6,12 ----

         DATA DIVISION.
         WORKING-STORAGE SECTION.
+        01  image-lib                         pic x(72) value spaces.

         01  END-OF-CHAIN                      PIC  S9(4)  COMP VALUE 15.
         01  END-OF-DATA-SET                   PIC  S9(4)  COMP VALUE 11.
***************
*** 48,53 ****
--- 49,57 ----

         PROCEDURE DIVISION.
         begin-here.
+
+            accept image-lib from environment "IMAGE_LIB".
+            if image-lib not = spaces then call image-lib.

             display "opening db..."


$ ccbl -Cp img01_acu.cob

$ export SHARED_LIBRARY_EXTENSION=.sl
$ export IMAGE_LIB=/opt/eloquence/8.0/lib/hpux32/libimage3k.sl

$ runcbl img01_acu
```

Module 2 – Admin Tasks

Lab 2.1 – Perform offline backup & recovery of your Eloquence instance

In this exercise you will perform an *offline* backup of your data and log volumes, intentionally damage the database(s) with dberase, and then recover your server instance from the backup.

Stop your Eloquence server instance. Optionally „flush and shrink“ your log volume file(s) with dblogreset. Backup data and log volume files using a tool of your choice, for example tar. It makes sense to include the eloqdb.cfg in the backup as well. Restart your server instance afterwards.

Simulate a damage or disaster by using dberase on a selected dataset or a whole database. Perform recovery by stopping the server instance, restoring the data and log volume files from your backup, and then restarting the recovered server instance. Use dbinfo to verify proper entry counts.

```
team1@unix:~> /etc/init.d/eloq8 stop team1    # /sbin/init.d/... for hp-ux

Stopping eloqdb[team1] daemon                                      done


team1@unix:~> ls -l db

-rw-------  1 team1 users 3670016 2005-11-25 00:35 data-01.vol
-rw-------  1 team1 users 2621440 2005-11-25 00:35 log-01.vol


team1@unix:~> dblogreset -help

ELOQUENCE DBLOGRESET (C) Copyright 2008 Marxmeier Software AG (B.08.00)
usage: dblogreset [options]
options:
 -v        - verbose
 -d flags  - debug flags
 -c cfg    - configuration file name
 -b size   - Buffer cache size (MB)


team1@unix:~> dblogreset -v -c eloqdb.cfg

ELOQUENCE DBLOGRESET (C) Copyright 2008 Marxmeier Software AG (B.08.00)
Reading configuration ...
Opening root volume
Initializing cache subsystem ...
Initializing subsystems ...
Recovering committed transactions ...
Resetting transaction log volume ...
Closing volume ...
done.


team1@unix:~> tar -cvf $HOME/backup1.tar elo*.cfg db/*.vol

eloqdb.cfg
db/data-01.vol
db/log-01.vol


team1@unix:~> /etc/init.d/eloq8 start team1    # /sbin/init.d/... for hp-ux

Starting eloqdb[team1] daemon                                      done
```

```
( now we „simulate“ a database disaster... )

team1@unix:~> dberase -help

ELOQUENCE DBERASE (C) Copyright 2008 Marxmeier Software AG (B.08.00)

usage: dberase [options] database [set ...]
options:
 -help   - show usage (this list)
 -u user - set user name
 -p pswd - set password
 -d flgs - debug flags


team1@unix:~> dberase toydb

ELOQUENCE DBERASE (C) Copyright 2008 Marxmeier Software AG (B.08.00)


team1@unix:~> dbinfo toydb

ELOQUENCE DBINFO (C) Copyright 2008 Marxmeier Software AG (B.08.00)

Processing database: toydb

SET NAME                RECLEN CAPACITY ENTRIES
---------------- --- - ------ -------- --------
PRODUCTS         001 M     30        0        0
CUSTOMERS        002 M    260        0        0
ORDER-MASTER     003 A      8        0        0
INVOICES         004 D     40        0        0
ORDERS           005 D     18        0        0
ORDER-DETAILS    006 D     20        0        0


team1@unix:~> /etc/init.d/eloq8 stop team1    # /sbin/init.d/... for hp-ux

Stopping eloqdb[team1] daemon                                      done


team1@unix:~> tar -xvf backup1.tar db

db/data-01.vol
db/log-01.vol


team1@unix:~> /etc/init.d/eloq8 start team1    # /sbin/init.d/... for hp-ux

Starting eloqdb[team1] daemon                                      done


team1@unix:~/tmp> dbinfo toydb

ELOQUENCE DBINFO (C) Copyright 2008 Marxmeier Software AG (B.08.00)

Processing database: toydb

SET NAME                RECLEN CAPACITY ENTRIES
---------------- --- - ------ -------- --------
PRODUCTS         001 M     30     1456       12
CUSTOMERS        002 M    260      229        8
ORDER-MASTER     003 A      8     1456       15
INVOICES         004 D     40     1149        1
ORDERS           005 D     18     1771       15
ORDER-DETAILS    006 D     20     1771       15
```

Lab 2.2 – Perform online backup & recovery of your Eloquence instance

In this exercise you will perform an *online* backup of your data volume files, intentionally damage the database server by deleting volume files with rm, and then recover your server instance from the backup. To show the backup „sync point“ you will also add entries before and during backup.

Use query3k to add some „before backup 2“ entry, for example to the PRODUCTS dataset. Switch the server instance to online backup mode with dbctl. Use query3k to add some „during backup 2“ entry to your database. Backup the data volumes, which are now „idle“, using tar or another tool of your choice. Finally use dbctl to disable online backup mode again.

Simulate a disaster by using rm to delete one or more database volume files. Notice that the server will continue to be operational as long as it keeps the volume files open, as the files have only been removed from the Unix directory. However, they will be fully deallocated as soon as the database server is shut down and closes those files.

After stopping your damaged instance, recover (all) the data volumes from your backup. If you deleted log volumes as part of your intentional database disaster, you need to recreate them with dbvolextend (matching the names in eloqdb.cfg). Restart your recovered server instance and use query3k to check the database contents. You will see the „before backup 2“ entry in your dataset, but the „during backup 2“ entry will not be there, as it was written after the „snapshot“ resulting from the „dbctl backup start“ command.

```
team1@unix:~> query3k

B.08.00.01 Eloquence QUERY3K  SUN, NOV 27, 2005, 10:42 AM
Copyright 2004-2008 Marxmeier Software AG
Copyright 2004 Hewlett-Packard Development Company, L.P.

>base=toydb
PASSWORD = >>
MODE = >>1

>add products
PRODUCT-NO      =>>Team1D
PRODUCT-NAME    =>>Before Backup 2
PRICE           =>>0
PRODUCT-LINE    =>>-
QUANTITY        =>>0

PRODUCT-NO      =>>//

>exit


team1@unix:~> dbctl -help

usage: dbctl [options] command [arg ...]
options:
 -help         - show usage (this list)
 -u name       - user name
 -p pswd       - password
 -h host       - host name or address
 -s service    - service name or port number
 -d flags      - debug flags
commands:
 help          - show list of available commands
 help command  - show usage of specific command
```

```
team1@unix:~> dbctl help

available commands:
 backup          dbstore         dbrestore       forwardlog
 shutdown        list            cancelthread    killthread
 logfile         logflags        syncmode        bimport
 replication     statfile        statfileflags   sessionstatfile
 sessionstatmode


team1@unix:~> dbctl help backup

usage: backup {START|STOP|STATUS}


team1@unix:~> dbctl backup status

On-line backup mode is inactive.


team1@unix:~> dbctl -u dba backup start

On-line backup mode has been started.


team1@unix:~> query3k

B.08.00.01 Eloquence QUERY3K  SUN, NOV 27, 2005, 10:46 AM
Copyright 2004-2008 Marxmeier Software AG
Copyright 2004 Hewlett-Packard Development Company, L.P.

>base=toydb
PASSWORD = >>
MODE = >>1

>add products
PRODUCT-NO      =>>Team1E
PRODUCT-NAME    =>>During Backup 2
PRICE           =>>1
PRODUCT-LINE    =>>X
QUANTITY        =>>0

PRODUCT-NO      =>>//

>exit


team1@unix:~> ll db

-rw-------  1 team1 users 3670016 2005-11-27 10:46 data-01.vol
-rw-------  1 team1 users 2621440 2005-11-27 10:46 log-01.vol


team1@unix:~> tar -cvf backup2.tar elo*.cfg db/data-*.vol

eloqdb.cfg
db/data-01.vol


team1@unix:~> dbctl -u dba backup stop

On-line backup mode has been stopped.


team1@unix:~> ls db/*.vol

db/data-01.vol  db/log-01.vol
```

```
( now we „simulate“ a database disaster... )

team1@unix:~> rm db/*.vol


team1@unix:~> /etc/init.d/eloq8 stop team1    # /sbin/init.d/... for hp-ux

Stopping eloqdb[team1] daemon                                         done


team1@unix:~> tar -xvf backup2.tar db

db/data-01.vol


( we still have missing log volume files... )

team1@unix:~> dbvoldump -c eloqdb.cfg

ELOQUENCE DBVOLDUMP (C) Copyright 2008 Marxmeier Software AG (B.08.00)

ID  Type Path
1   DATA /home/team1/db/data-01.vol
2   LOG  [missing]

ID  Type  Cur.Sz  Ext.Sz  Max.Sz    Free    Used
1   DATA     3.5     1.0    50.0     1.8     1.7


team1@unix:~> dbvolextend -help

ELOQUENCE DBVOLEXTEND (C) Copyright 2008 Marxmeier Software AG (B.08.00)

usage: dbvolextend [options] volume_file_name
       dbvolextend [options] -R
options:
 -R         - recreate missing LOG volumes if possible
 -t type    - Volume type (DATA, LOG)
 -v         - verbose
 -d flags   - debug flags
 -c cfg     - configuration file name
 -s sz      - Initial size
 -e sz      - Extension size
 -m sz      - Max volume size


team1@unix:~> dbvolextend -c eloqdb.cfg -v -R

ELOQUENCE DBVOLEXTEND (C) Copyright 2008 Marxmeier Software AG (B.08.00)

Reading configuration ...
Opening root volume
Restoring log volume /home/team1/db/log-01.vol
Closing volume ...
done.


team1@unix:~> dbvoldump -c eloqdb.cfg

ELOQUENCE DBVOLDUMP (C) Copyright 2008 Marxmeier Software AG (B.08.00)

ID  Type Path
1   DATA /home/team1/db/data-01.vol
2   LOG  /home/team1/db/log-01.vol

ID  Type  Cur.Sz  Ext.Sz  Max.Sz    Free    Used
1   DATA     3.5     1.0    50.0     1.8     1.7
2   LOG      2.5     1.0     0.0     2.5     0.0
```

```
team1@unix:~> /etc/init.d/eloq8 start team1    # /sbin/init.d/... for hp-ux

Starting eloqdb[team1] daemon                                      done


team1@unix:~> dbinfo toydb

ELOQUENCE DBINFO (C) Copyright 2008 Marxmeier Software AG (B.08.00)

Processing database: toydb

SET NAME               RECLEN CAPACITY ENTRIES
---------------- --- - ------ -------- --------
PRODUCTS         001 M     30     1456       13
CUSTOMERS        002 M    260      229        8
ORDER-MASTER     003 A      8     1456       15
INVOICES         004 D     40     1149        1
ORDERS           005 D     18     1771       15
ORDER-DETAILS    006 D     20     1771       15


team1@unix:~> query3k

B.08.00.01 Eloquence QUERY3K  SUN, NOV 27, 2005, 10:59 AM
Copyright 2004-2008 Marxmeier Software AG
Copyright 2004 Hewlett-Packard Development Company, L.P.

>base=toydb
PASSWORD = >>
MODE = >>5

>list products

PRODUC  PRODUCT-NAME         PRICE  PR  QUANTI

A00003  POKER DICE SET         125  10     500
Team1A  Fun Product            399  X        1
A00008  POSTER PAINTS           95  30    1250
A00009  COLOURING BOOK          65  30    1000
A00001  PACK OF CARDS           75  10    1500
A00010  ERASER GIFT SET        185  30    1500
A00002  LUDO SET              1250  10     750
A00005  15" PINK RABBIT       1745  20     200
A00007  SET OF CRAYONS         175  30     500
A00004  12" TEDDY BEAR        1525  20     250
A00006  SET OF PANDAS         2500  20     150
Team1B  Now on Unix            199  U        2
Team1D  Before Backup 2          0  -        0

>exit
```

Lab 2.3 – Setup forward logging and perform recovery of your instance

In this exercise you will configure forward logging for your server instance, perform an online backup, as well as apply database changes before, during and after this backup. You will damage your database intentionally and then recover it from the backup and replay the log transactions.

Change your eloqdb.cfg file to enable forward logging. Use the %N pattern to get dynamically generated logfile names. Restart the server instance to activate the config change. Use query3k to add some „before backup 3“ entry to your PRODUCTS dataset. Switch to online backup mode using dbctl. Use query3k to add another entry „during backup 3“ to your dataset. Backup the data volume file(s) with tar. Switch off online backup mode with dbutil. Add yet another entry „after backup 3“ using query3k.

Before damaging your database intentionally, record the current date and time. Now use dberase against the PRODUCTS dataset to „simulate“ a database disaster. By using the recorded date and time, you will be able to recover the forward logs up the point right before the „artificial disaster“, which is important because the dberase transactions are also contained in the forward logfile.

Note that in real-life you would probably use messages from the eloqdb server log and/or database auditing info from the forward logs to find or estimate the time when the “data incident” happened. You will learn more about those two methods in subsequent labs of this module.

After damaging your database, stop the server instance and recover the data volume files from your backup. Apply the forward logfiles (up to, but not including the dberase) on top of the restored data volumes before restarting the server instance. Do NOT attempt to re-start your eloqdb server before applying the forward logs! (this would increment the logfile generation count and make dbrecover impossible without another restore of the volume files).

Move away any “leftover” forwardlog files after dbrecover, before starting the new server process.

After the recovered instance has been restarted, verify the proper dataset contents with query3k.

```
team1@unix:~> cp -p eloqdb.cfg eloqdb.old

team1@unix:~> ## customize eloqdb.cfg (see diff output below for changes)


team1@unix:~> diff eloqdb.old eloqdb.cfg

479c479
< #FwLog = /data/fwlog/fw-%N.log
---
> FwLog = /home/team1/log/fw-%N.log

team1@unix:~> mkdir log


team1@unix:~> /etc/init.d/eloq8 restart team1    # /sbin/init.d/... for hp-ux

Stopping eloqdb[team1] daemon                                      done
Starting eloqdb[team1] daemon                                      done


team1@unix:~> ll log

-rw-------  1 team1 users 118 2005-11-27 17:40 fw-1-1.log
```

```
team1@unix:~> dbctl forwardlog status

Forward-logging is enabled.
Forward-log is '/home/team1/log/fw-1-1.log'.


team1@unix:~> query3k

B.08.00.01 Eloquence QUERY3K  SUN, NOV 27, 2005,  5:44 PM
Copyright 2004-2008 Marxmeier Software AG
Copyright 2004 Hewlett-Packard Development Company, L.P.

>base=toydb
PASSWORD = >>
MODE = >>1

>add products
PRODUCT-NO      =>>Team1E
PRODUCT-NAME    =>>Before Backup L
PRICE           =>>0
PRODUCT-LINE    =>>L
QUANTITY        =>>0

PRODUCT-NO      =>>//

>exit


team1@unix:~> dbctl -u dba backup start

On-line backup mode has been started.


( notice that backup start switches to a new fw log generation... )

team1@unix:~> ll log

-rw-------  1 team1 users   724 2005-11-27 17:48 fw-1-1.log
-rw-------  1 team1 users   227 2005-11-27 17:48 fw-2-1.log


team1@unix:~> query3k

B.08.00.01 Eloquence QUERY3K  SUN, NOV 27, 2005,  5:48 PM
Copyright 2004-2008 Marxmeier Software AG
Copyright 2004 Hewlett-Packard Development Company, L.P.

>base=toydb
PASSWORD = >>
MODE = >>1

>add products
PRODUCT-NO      =>>Team1F
PRODUCT-NAME    =>>During Backup L
PRICE           =>>0
PRODUCT-LINE    =>>L
QUANTITY        =>>0

PRODUCT-NO      =>>//

>exit


team1@unix:~> tar -cvf backup3.tar elo*.cfg db/data-*.vol

eloqdb.cfg
db/data-01.vol


team1@unix:~> dbctl -u dba backup stop

On-line backup mode has been stopped.
```

```
team1@unix:~> ll log

-rw-------  1 team1 users   724 2005-11-27 17:48 fw-1-1.log
-rw-------  1 team1 users   793 2005-11-27 17:51 fw-2-1.log


team1@unix:~> query3k

B.08.00.01 Eloquence QUERY3K  SUN, NOV 27, 2005,  5:52 PM
Copyright 2004-2008 Marxmeier Software AG
Copyright 2004 Hewlett-Packard Development Company, L.P.

>base=toydb
PASSWORD = >>
MODE = >>1

>add products
PRODUCT-NO      =>>Team1G
PRODUCT-NAME    =>>After Backup L
PRICE           =>>0
PRODUCT-LINE    =>>L
QUANTITY        =>>0

PRODUCT-NO      =>>//

>exit


( now prepare the intentional database damage... )

team1@unix:~> date

Sun Nov 27 17:54:04 CET 2005


team1@unix:~> dberase toydb products

ELOQUENCE DBERASE (C) Copyright 2008 Marxmeier Software AG (B.08.00)


team1@unix:~> dbinfo toydb products

ELOQUENCE DBINFO (C) Copyright 2008 Marxmeier Software AG (B.08.00)

Processing database: toydb

SET NAME               RECLEN CAPACITY ENTRIES
---------------- --- - ------ -------- --------
PRODUCTS         001 M     30        0        0


team1@unix:~> /etc/init.d/eloq8 stop team1    # /sbin/init.d/... for hp-ux

Stopping eloqdb[team1] daemon


team1@unix:~> ll db

-rw-------  1 team1 users 3670016 2005-11-27 17:55 data-01.vol
-rw-------  1 team1 users 2621440 2005-11-27 17:55 log-01.vol

team1@unix:~> ll log

-rw-------  1 team1 users   724 2005-11-27 17:48 fw-1-1.log
-rw-------  1 team1 users 13167 2005-11-27 17:55 fw-2-1.log


team1@unix:~> tar -xvf backup3.tar db

db/data-01.vol
```

```
team1@unix:~> dbrecover -help

ELOQUENCE DBRECOVER (C) Copyright 2008 Marxmeier Software AG (B.08.00)
usage: dbrecover [options]
options:
 -t tmpdir    - directory to be used for temporary files
 -v           - verbose
 -d flags     - debug flags
 -c cfg       - configuration file name
 -b size      - Buffer cache size (MB)
 -T timestamp - recover until point in time (incl.)

timestamp formats:
 YYYY-MM-DD HH:MM:SS
 MM/DD/YYYY HH:MM:SS
 DD.MM.YYYY HH:MM:SS
 note: any character may be used to separate date and time
       time part is optional (defaults to 00:00:00)


team1@unix:~> dbrecover -c eloqdb.cfg -T '2005-11-27 17:54' -v

ELOQUENCE DBRECOVER (C) Copyright 2008 Marxmeier Software AG (B.08.00)
Opening root volume
Recovering from forward-log …
R1: processing forward-log file: '/home/team1/log/fw-2-1.log'
14 actions have been successfully recovered.
Database environment is now up-to-date until Sun Nov 27 17:53:46 2005.
done.

team1@unix:~> mkdir log/done; mv log/fw*.log log/done  # move “leftover” logs away

team1@unix:~> /etc/init.d/eloq8 start team1    # /sbin/init.d/... for hp-ux

Starting eloqdb[team1] daemon                                      done


team1@unix:~> query3k

B.08.00.01 Eloquence QUERY3K  SUN, NOV 27, 2005,  6:00 PM
Copyright 2004-2008 Marxmeier Software AG
Copyright 2004 Hewlett-Packard Development Company, L.P.

>base=toydb
PASSWORD = >>
MODE = >>5

>list products

PRODUC  PRODUCT-NAME         PRICE  PR  QUANTI

A00003  POKER DICE SET         125  10     500
Team1A  Fun Product            399  X        1
A00008  POSTER PAINTS           95  30    1250
A00009  COLOURING BOOK          65  30    1000
A00001  PACK OF CARDS           75  10    1500
A00010  ERASER GIFT SET        185  30    1500
A00002  LUDO SET              1250  10     750
A00005  15" PINK RABBIT       1745  20     200
A00007  SET OF CRAYONS         175  30     500
A00004  12" TEDDY BEAR        1525  20     250
A00006  SET OF PANDAS         2500  20     150
Team1B  Now on Unix            199  U        2
Team1D  Before Backup 2          0  -        0
Team1E  Before Backup L          0  L        0
Team1F  During Backup L          0  L        0
Team1G  After Backup L           0  L        0

>exit
```

Lab 2.4 – Use tools and logs for Monitoring and Troubleshooting

This exercise has two parts. In the first one, we will use dbvoldump, dbctl and http to monitor status information like volume free space or active database connections. In the second one, we will look at client and server logfiles for a simulated error condition.

Step (a): Status monitoring with dbvoldump, dbctl and http

Use dbvoldump to examine the data and log volumes of your server instance and their respective disk space usage. By default, the volume files grow dynamically, but you can optionally allocate them with their full size, if you want to protect yourself from being surprised by shortages in disk space at the file system level over time. If your existing volume files run short of available free space, you need to add one or more with dbvolextend.

```
team1@unix:~> dbvoldump -c eloqdb.cfg

ELOQUENCE DBVOLDUMP (C) Copyright 2008 Marxmeier Software AG (B.08.00)

ID  Type Path
1   DATA /home/team1/db/data-01.vol
2   LOG  /home/team1/db/log-01.vol

ID  Type  Cur.Sz  Ext.Sz  Max.Sz    Free    Used
1   DATA     3.5     1.0     0.0     1.7     1.8
2   LOG      2.5     1.0     0.0     2.5     0.0
```

To examine information about active database connections, we will use a small program (img02) that opens the database and locks a dataset for 60 seconds. By using three Unix sessions, we can invoke two concurrent instances of this program (with a slight time offset) and then use dbctl to display information about active client sessions, open databases, and active or pending locks. We could use dbctl commands like cancelthread or killthread, if needed.

The session snippets below mark the 3 concurrent sessions with an „A:“ or „B:“ or „C:“ prefix.

```
A:  team1@unix:~> dbctl -help
A:
A:  usage: dbctl [options] command [arg ...]
A:  options:
A:   -help        - show usage (this list)
A:   -u name      - user name
A:   -p pswd      - password
A:   -h host      - host name or address
A:   -s service   - service name or port number
A:   -d flags     - debug flags
A:  commands:
A:   help         - show list of available commands
A:   help command - show usage of specific command
A:
A:
A:  team1@unix:~> dbctl help
A:
A:  available commands:
A:   backup          dbstore         dbrestore       forwardlog
A:   shutdown        list            cancelthread    killthread
A:   logfile         logflags        syncmode        bimport
A:   replication     statfile        statfileflags   sessionstatfile
A:   sessionstatmode
A:
A:
```

```
A:  team1@unix:~> dbctl help list
A:
A:  list arguments:
A:   session         dbopen          db              lock
A:   thread
A:  usage: list type [/NOTITLE|/COUNT] [argument] [filter ...]


( session B will lock a dataset for 60 seconds... )

B:  team1@unix:~> ls -l tmp/img02*
B:
B:  -rwxr-xr-x  1 team1 users 10242 2006-01-03 23:35 tmp/img02
B:  -rw-r--r--  1 team1 users  2757 2005-11-30 00:06 tmp/img02.c
B:  -rw-r--r--  1 team1 users  2540 2005-11-30 00:06 tmp/img02.o
B:
B:  team1@unix:~> tmp/img02
B:
B:  opening db...
B:  locking products...
B:  waiting 60 secs...


( session C blocks attempting locking the same set... )

C:  team1@unix:~> tmp/img02
C:
C:  opening db...
C:  locking products...


( looking at the resulting status from session A again... )

A:  team1@unix:~> dbctl list session
A:
A:  TID  IP ADDR              User / Login
A:  ---- -------------------- -------------------------------------------
A:     8 127.0.0.1:20608      team1 / public
A:       uid{1001}pid{6330}pname{tmp/img02}
A:     9 127.0.0.1:20609      team1 / public
A:       uid{1001}pid{6350}pname{tmp/img02}
A:    10 127.0.0.1:20611      team1 / public
A:       uid{1001}pid{6382}pname{dbctl list session}
A:
A:
A:  team1@unix:~> dbctl list db
A:
A:  Database         Ref  WrShrd WrExcl RdShrd RdExcl WrSngl
A:  ---------------- ---- ------ ------ ------ ------ ------
A:  TOYDB               2      2      0      0      0      0
A:
A:
A:  team1@unix:~> dbctl list dbopen toydb
A:
A:  TID  IP ADDR              M   User / Login
A:  ---- -------------------- --- -------------------------------------------
A:     8 127.0.0.1:20608      1   team1 / public
A:       uid{1001}pid{6330}pname{tmp/img02}
A:     9 127.0.0.1:20609      1   team1 / public
A:       uid{1001}pid{6350}pname{tmp/img02}
A:
A:
A:  team1@unix:~> dbctl list lock
A:
A:  TID  Database         DBID Status  Mode Qualifier
A:  ---- ---------------- ---- ------- ---- ---------------
A:     8 TOYDB               1 GRANTED    4 set 1
A:     9 TOYDB               1 BLOCKED    4 set 1
```

```
( when the program in session B releases the lock, we see... )

B:  unlocking products...
B:  closing db...
B:
B:  team1@unix:~>


C:  waiting 60 secs...


A:  team1@unix:~> dbctl list lock
A:
A:  TID  Database         DBID Status  Mode Qualifier
A:  ---- ---------------- ---- ------- ---- ---------------
A:     9 TOYDB               1 GRANTED    4 set 1
```

By enabling the http status display in your eloqdb.cfg file, you can also view the information seen in dbvoldump and dbctl from a web browser. Modify your config file using the HTTP port number assigned by the teacher (as each team will have to use a different one) and restart your instance to activate the change. View the status information from a web browser while repeating the concurrent img02 invocations like done above.

```
team1@unix:~> more /etc/services

 (...)

http_team1  3081/tcp  # team1 db server
http_team2  3082/tcp  # team2 db server
http_team3  3083/tcp  # team3 db server
http_team4  3084/tcp  # team4 db server

 (...)


team1@unix:~> cp -p eloqdb.cfg eloqdb.old

team1@unix:~> ## customize eloqdb.cfg (see diff output below for changes)


team1@unix:~> diff eloqdb.old eloqdb.cfg

42c42
< #ServiceHttp =
---
> ServiceHttp = http_team1


team1@unix:~> /etc/init.d/eloq8 restart team1    # /sbin/init.d/... for hp-ux

Stopping eloqdb[team1] daemon                                       done
Starting eloqdb[team1] daemon                                       done


team1@unix:~> netstat -a | grep _team1

tcp        0      0 *:eqdb_team1            *:*               LISTEN
tcp        0      0 *:http_team1            *:*               LISTEN


(now browse http://your-host-ip:3081 to view the http status display)
```

Step (b): Using client and server logfiles for troubleshooting

In this step we use a „database alias“ environment variable to cause a DBOPEN failure in the sample program used in step (a) already. The program will call DBEXPLAIN to report the error condition. The DBEXPLAIN output typically looks similar to MPE/iX, but may also include secondary status codes specific to Eloquence. The server side logfile does not contain any helpful messages regarding the error as we set up our instance with log flags „*0E1“ only.

To examine details of the error condition, we enable client side logging by the Image3K libary and also increase the server side logging details with dbctl (using log flags „*1E2“ in this case). Both logs contain hints regarding the incorrect database name passed to the DBOPEN invocations. After looking at the logs, remember to disable the client side logging and return to the original log flags for the server side logging. You should also delete the offending „database alias“ environment var.

```
team1@unix:~> export EQ3K__TOYDB=toydb.oops

team1@unix:~> tmp/img02

opening db...
IMAGE STATUS -1, OP 401(1)
ffff 0000 0034 0000 0000 0191 ffff 0000 0001 0401
DBOPEN(1): Unable to open database [-1]


team1@unix:~> tail eloqdb.log

** Wed Jan  4 11:51:48 2006   Flags = [*0E1]

Wed 04 11:51:48 ( 7112) D0: Server patch level PE80-0906080
 ...
Wed 04 11:51:48 ( 7112) D0: Eloquence database server active
Wed 04 11:53:43 ( 7112) E1: [9] Login: "public" from 127.0.0.1/team1


team1@unix:~> export EQ3K_DEBUG=client.log

team1@unix:~> dbctl -u dba logflags "*1E2"

Log flags have been set to "*1E2".


team1@unix:~> tmp/img02

opening db...
IMAGE STATUS -1, OP 401(1)
ffff 0000 0034 0000 0000 0191 ffff 0000 0001 0401
DBOPEN(1): Unable to open database [-1]


team1@unix:~> tail client.log

Wed 04 11:55:35 ( 7187): *** start ***
Wed 04 11:55:35 ( 7187): IMAGE3K B.08.00.00
Wed 04 11:55:35 ( 7187): ELOQDB  B.08.00.02
Wed 04 11:55:35 ( 7187): dbopen: mode=1
Wed 04 11:55:35 ( 7187): dbopen: dbname=TOYDB
Wed 04 11:55:35 ( 7187): dbopen: dbname=toydb.oops
Wed 04 11:55:35 ( 7187): dbopen: user=0 pswd=7
Wed 04 11:55:35 ( 7187): hp3k__map_status: status=-1
```

```
team1@unix:~> tail eloqdb.log

** Wed Jan  4 11:55:14 2006   Flags = [*1E2]

Wed 04 11:55:14 ( 7112) D1: [9] Disconnected 127.0.0.1:41629 (TID 9)
Wed 04 11:55:35 ( 7112) D1: [2] Connection from 127.0.0.1:41630 (TID 9)
Wed 04 11:55:35 ( 7112) E2: [9] Audit: protocol{9}os{Linux}ip{127.0.0.1}user{team1}
Wed 04 11:55:35 ( 7112) E2: [9] Audit: uid{1001}pid{7187}pname{tmp/img02}
Wed 04 11:55:35 ( 7112) E1: [9] Login: "public" from 127.0.0.1:41630/team1
Wed 04 11:55:35 ( 7112) I1: [9] Database toydb.oops does not exist
Wed 04 11:55:35 ( 7112) D1: [9] Disconnected 127.0.0.1:41630 (TID 9)

team1@unix:~> unset EQ3K_DEBUG

team1@unix:~> dbctl -u dba logflags "*0E1"

Log flags have been set to "*0E1".


team1@unix:~> unset EQ3K__TOYDB
```

As an optional step, you might try different log flags (for example „*1E3P2“) to get more detailed messages for the error context. You might also try taking detailed client and server side logs when the sample program runs successfully.

Lab 2.5 – Perform structural database changes with dbutil

In this exercise you will perform several structural database modifications with dbutil. The lab is divided into 3 parts. You will first create a new item and add a respective field to an existing set. You will then create a new detail set with several (new) fields and link a path to an existing master set. You will finally define and add an index item to this detail set. This is a feature that does not exist in TurboImage on MPE/iX without third party indexing tools.

Step (a): Create new item and add field to existing dataset

Better backup your database instance before performing the restructuring (in case something does not work as expected during the remainder of this excercise); examine the current database structure with prschema to familiarize with the starting point; then create a dbutil script to define a new item VENDOR-NAME of type X20 and add a respective field to the existing PRODUCTS dataset.

Use „dbutil -n" to test your change script before applying the modifications. Verify the modified database structure with tools like dbtables, prschema, or query3k. Also use query3k to add (update) some test data in the new column.

```
team1@unix:~> dbctl -u dba backup start

On-line backup mode has been started.

team1@unix:~> tar -cvf backup4.tar db/data*.vol

db/data-01.vol

team1@unix:~> dbctl -u dba backup stop

On-line backup mode has been stopped.


team1@unix:~> prschema -T toydb

ELOQUENCE PRSCHEMA (C) Copyright 2008 Marxmeier Software AG (B.08.00)

# Schema definition for data base TOYDB
# TurboIMAGE compatible schema file, Wed Nov 30 22:38:37 2005 CET

BEGIN DATA BASE TOYDB;

PASSWORDS:
        1 READING;
        2 WRITING;

ITEMS:
         (...)
        PRICE,                P8;
        PRODUCT-LINE,         X2;
        PRODUCT-NAME,         X16;
        PRODUCT-NO,           X6;
        QUANTITY,             I;
         (...)

SETS:

N:      PRODUCTS, M (1/2);
E:      PRODUCT-NO (1),
        PRODUCT-NAME,
        PRICE,
        PRODUCT-LINE,
        QUANTITY;
C:      1456;

 (...)

END.


team1@unix:~> ## create a script with dbutil commands (see below for contents)

team1@unix:~> cat change-1.txt

  database "toydb";

  create item
    vendor-name, X20;

  change set products
    add item vendor-name;

  exit;


team1@unix:~> dbutil -help

ELOQUENCE DBUTIL (C) Copyright 2008 Marxmeier Software AG (B.08.00)
usage: dbutil [options] [file|-]
options:
 -help      - show usage (this list)
 -u name    - user name (default "dba")
 -p pswd    - password
 -h host    - host name or ip address (and service)
 -s service - service name or port number
 -n         - pretend          (batch mode only)
 -v         - verbose          (batch mode only)
 -e cnt     - abort processing after encountering cnt errors
 -d flgs    - debug flags
 -T         - HP3000 TurboImage compatibility mode

If a file is specified, dbutil will process in batch mode
and process any statements in the batch file.
If the file argument is not specified dbutil runs in
interactive mode.
If the -n option is present, no changes will be made to the
database. Processing will end after checking the input file
and the analyse phase.


team1@unix:~> dbutil -v -T -n change-1.txt

ELOQUENCE DBUTIL (C) Copyright 2008 Marxmeier Software AG (B.08.00)
Processing specification ...

Checking database consistency ...
Consistency check completed successfully

Database restructure analysis:
 PRODUCTS
  * Record reorganized due to new item
Data restructure process required.

Uploading modified schema ...
*** Test mode is active, discarding all changes
```

```
team1@unix:~> dbutil -v -T change-1.txt

ELOQUENCE DBUTIL (C) Copyright 2008 Marxmeier Software AG (B.08.00)
Processing specification ...

Checking database consistency ...
Consistency check completed successfully

Database restructure analysis:
 PRODUCTS
  * Record reorganized due to new item
Data restructure process required.

Uploading modified schema ...
Restructuring database ...
done


team1@unix:~> dbtables -s toydb

ELOQUENCE DBTABLES (C) Copyright 2008 Marxmeier Software AG (B.08.00)

                         D A T A   S E T   P A T H E S

Data Set Name    Item Name         Data Set Name    Set  Item Name         Item
                                                    Num                    Num

 (...)

                         D A T A   S E T   F I E L D S

Data Set Name    Set  Item Name        Item  Item  Recd   Item  Item  Item
                 Num                   Num   Type  Ofs    Cnt   Len   Role

PRODUCTS           1  PRODUCT-NO         15  X6       0     1     6  Srch
                      PRODUCT-NAME       14  X16      6     1    16
                      PRICE              12  P4      22     1     4
                      PRODUCT-LINE       13  X2      26     1     2
                      QUANTITY           16  I2      28     1     2
                      VENDOR-NAME        24  X20     30     1    20

 (...)


team1@unix:~> query3k

B.08.00.01 Eloquence QUERY3K  WED, NOV 30, 2005, 10:40 PM
Copyright 2004-2008 Marxmeier Software AG
Copyright 2004 Hewlett-Packard Development Company, L.P.

>base=toydb
PASSWORD = >>
MODE = >>1

>form products

DATA BASE: TOYDB                        WED, NOV 30, 2005, 10:40 PM

DATA BASE LANGUAGE ATTRIBUTE: NATIVE-3000

SET NAME:
   PRODUCTS,MANUAL

      ITEMS:
         PRODUCT-NO,           X6            <<KEY ITEM>>
         PRODUCT-NAME,         X16
         PRICE,                P8
         PRODUCT-LINE,         X2
         QUANTITY,             I1
         VENDOR-NAME,          X20

      CAPACITY: 2016           ENTRIES: 16
```

```
>find products.price=0
USING SERIAL READ
4  ENTRIES QUALIFIED

>replace vendor-name="test only"; end

>list products

PRODUC  PRODUCT-NAME         PRICE  PR  QUANTI  VENDOR-NAME

A00003  POKER DICE SET         125  10     500
Team1A  Fun Product            399  X        1
A00008  POSTER PAINTS           95  30    1250
A00009  COLOURING BOOK          65  30    1000
A00001  PACK OF CARDS           75  10    1500
A00010  ERASER GIFT SET        185  30    1500
A00002  LUDO SET              1250  10     750
A00005  15" PINK RABBIT       1745  20     200
A00007  SET OF CRAYONS         175  30     500
A00004  12" TEDDY BEAR        1525  20     250
A00006  SET OF PANDAS         2500  20     150
Team1B  Now on Unix            199  U        2
Team1D  Before Backup 2          0  -        0  test only
Team1E  Before Backup L          0  L        0  test only
Team1F  During Backup L          0  L        0  test only
Team1G  After Backup L           0  L        0  test only

>exit
```

Note: You can no longer use the IMG01 program against the modified database, since it reads the PRODUCTS dataset using the „@“ item list and thus needs at least an updated DBGET buffer size. If you like, you can adapt and recompile the source code as an optional exercise. However, you can also use the prepared version supplied as img03.c (or img03.cob respectively) or use diff to display the source code changes.

Step (b): Create new detail set with several fields and path to existing master set

Use another dbutil script for this step; create the new items OLD-PRICE, NEW-PRICE (same type as the existing item PRICE) as well as CHG-DATE (type X8); create a new detail set PRICE-HIST using PRODUCT-NO and the new items as fields; define a path to link this new detail set to the PRODUCTS master set via the PRODUCT-NO field; define this chain as sorted by CHG-DATE.

You will also need to grant write privileges to the new dataset, so that it becomes accessible to user class „writing“ of the database. Use dbutil to test and then apply your modifications. Verify results with dbtables, prschema or query3k. Also add some test data to the new detail set using query3k.

```
team1@unix:~> ## create a script with dbutil commands (see below for contents)

team1@unix:~> cat change-2.txt

  database "toydb";

  create item {
    old-price, P8;
    new-price, P8;
    chg-date,  X8;
  }

  create set price-hist, detail {
    add item
      product-no, old-price, new-price, chg-date;
    add path
      product-no (products (chg-date));
  }

  grant write on price-hist to "writing";

  exit;


team1@unix:~> dbutil -n -T -v change-2.txt

ELOQUENCE DBUTIL (C) Copyright 2008 Marxmeier Software AG (B.08.00)
Processing specification ...

Checking database consistency ...
Consistency check completed successfully

Database restructure analysis:
 PRODUCTS
  * Record reorganized due to new path
 PRICE-HIST
  * New data set
Data restructure process required.

Uploading modified schema ...
*** Test mode is active, discarding all changes
```

```
team1@unix:~> dbutil -T -v change-2.txt

ELOQUENCE DBUTIL (C) Copyright 2008 Marxmeier Software AG (B.08.00)
Processing specification ...

Checking database consistency ...
Consistency check completed successfully

Database restructure analysis:
 PRODUCTS
  * Record reorganized due to new path
 PRICE-HIST
  * New data set
Data restructure process required.

Uploading modified schema ...
Restructuring database ...
done


team1@unix:~> query3k

B.08.00.01 Eloquence QUERY3K  WED, NOV 30, 2005, 10:43 PM
Copyright 2004-2008 Marxmeier Software AG
Copyright 2004 Hewlett-Packard Development Company, L.P.

>base=toydb
PASSWORD = >>
MODE = >>1

>form sets

DATA BASE: TOYDB                        WED, NOV 30, 2005, 10:43 PM

DATA BASE LANGUAGE ATTRIBUTE: NATIVE-3000

                            ITEM   CURRENT   ENTRY    ENTRY   BLOCKING
SETS:                 TYPE  COUNT  CAPACITY  COUNT    LENGTH  FACTOR

   PRODUCTS           M     6      2553      16       25      0
   CUSTOMERS          M     10     229       8        130     0
   ORDER-MASTER       A     1      1456      15       4       0
   INVOICES           D     7      1149      1        20      0
   ORDERS             D     3      1771      15       9       0
   ORDER-DETAILS      D     4      1771      15       10      0
   PRICE-HIST         D     4      0         0        11      0


>form price-hist

DATA BASE: TOYDB                        WED, NOV 30, 2005, 10:44 PM

DATA BASE LANGUAGE ATTRIBUTE: NATIVE-3000

SET NAME:
   PRICE-HIST,DETAIL

      ITEMS:
         PRODUCT-NO,          X6            <<SEARCH ITEM>>
         OLD-PRICE,           P8
         NEW-PRICE,           P8
         CHG-DATE,            X8            <<SORT ITEM>>

      CAPACITY: 0              ENTRIES: 0
```

```
>list products

PRODUC  PRODUCT-NAME         PRICE  PR  QUANTI  VENDOR-NAME

         (...)
Team1A  Fun Product            399  X        1
         (...)

>add price-hist
PRODUCT-NO      =>>Team1A
OLD-PRICE       =>>399
NEW-PRICE       =>>259
CHG-DATE        =>>20051130

PRODUCT-NO      =>>//

>find products.product-no="Team1A"
1  ENTRIES QUALIFIED

>replace price="259"; end


>exit
```

Step (c): Create new index item for a field of the new detail set

Use another dbutil script to create a new index item CHG-DATE-IDX based on (the whole length of) CHG-DATE and add this index item to the detail set PRICE-HIST that you created in step (b). Apply the structure changes with dbutil and verify them with dbtables, prschema or query3k.

Use the FIND command inside query3k with the fields CHG-DATE versus CHG-DATE-IDX to confirm that the former performs serial access, whereas the latter uses indexed access (the query3k program displays a related message during the find command).

```
team1@unix:~> ## create a script with dbutil commands (see below for contents)

team1@unix:~> cat change-3.txt

  database "toydb";

  create iitem
    chg-date-idx = chg-date;

  change set price-hist
    add index chg-date-idx;

  exit;


team1@unix:~> dbutil -T -v change-3.txt

ELOQUENCE DBUTIL (C) Copyright 2008 Marxmeier Software AG (B.08.00)
Processing specification ...

Checking database consistency ...
Consistency check completed successfully

Database restructure analysis:
 PRICE-HIST
  * Record reorganized due to adding first index
  * New index added
Data restructure process required.

Uploading modified schema ...
Restructuring database ...
done


( note that the new schema is now incompatible with MPE/iX TurboImage... )


team1@unix:~> prschema -T toydb

ELOQUENCE PRSCHEMA (C) Copyright 2008 Marxmeier Software AG (B.08.00)

 (...)

BEGIN DATA BASE TOYDB;

 (...)

ITEMS:
        (...)
        VENDOR-NAME,         X20;
        OLD-PRICE,           P8;
        NEW-PRICE,           P8;
        CHG-DATE,            X8;

IITEMS:
        CHG-DATE-IDX      = CHG-DATE;
```

```
SETS:

 (...)

N:      PRICE-HIST, D (/2);
E:      PRODUCT-NO (!PRODUCTS(CHG-DATE)),
        OLD-PRICE,
        NEW-PRICE,
        CHG-DATE;
I:      CHG-DATE-IDX;
C:      3542;

END.


team1@unix:~> query3k

B.08.00.01 Eloquence QUERY3K  WED, NOV 30, 2005, 10:47 PM
Copyright 2004-2008 Marxmeier Software AG
Copyright 2004 Hewlett-Packard Development Company, L.P.

>base=toydb
PASSWORD = >>
MODE = >>5


>form price-hist

DATA BASE: TOYDB                        WED, NOV 30, 2005, 10:47 PM

DATA BASE LANGUAGE ATTRIBUTE: NATIVE-3000

SET NAME:
   PRICE-HIST,DETAIL

      ITEMS:
         PRODUCT-NO,            X6            <<SEARCH ITEM>>
         OLD-PRICE,             P8
         NEW-PRICE,             P8
         CHG-DATE,              X8            <<SORT ITEM>>

      TPI INDEXES:
         CHG-DATE-IDX,          G8

      CAPACITY: 3542             ENTRIES: 1


>FIND price-hist.chg-date="20051130"

USING SERIAL READ
1  ENTRIES QUALIFIED

>FIND price-hist.chg-date-idx="20051130"

1  ENTRIES QUALIFIED

>exit
```

Lab 2.6 – Perform database security changes with dbutil

Until now, we have used our database with the default security settings that have been established by Eloquence at database creation time. This basically grants full admin and data access to all users without any need for passwords. In this excercise we will customize this to tighter security.

When creating the database, Eloquence also created two default users (dba and public), with no passwords, as well as two associated database access groups (dba and public), which provide admin or connect privileges, respectively. For each TurboImage „user class“ in the schema definition, it created an additional database access group, named by the respective „user class“ (also known as „image password“) and having the user „public“ as member.

To customize this setup, you will first assign a password to the dba user and then create another password-protected user for your team. The latter should have connect privileges and be a member of the TOYDB-specific „user class“ called „WRITING“ (see database schema) to have read and write access to all the datasets. You will finally remove the user „public“ from the „READING“ and „WRITING“ groups, and thus limit database access to the password-protected user(s) only.

After these changes, you need to specify user and password information to the database tools and programs by command-line options or environment vars EQ_DBUSER and EQ_DBPASSWORD, respectively.

```
team1@unix:~> ## create a script with dbutil commands (see below for contents)

team1@unix:~> cat change-s.txt

  change user "dba" password "dilbert";

  create user "team1" password "dogbert";

  database "toydb";

  grant "writing" to "team1";

  revoke "reading", "writing" from "public";

  exit;


team1@unix:~> dbutil -Tv change-s.txt

ELOQUENCE DBUTIL (C) Copyright 2008 Marxmeier Software AG (B.08.00)
Processing specification ...

done


team1@unix:~> dbinfo toydb

ELOQUENCE DBINFO (C) Copyright 2008 Marxmeier Software AG (B.08.00)

Processing database: toydb

Fatal error #-21 while opening database
DBOPEN(9): Bad parameter value [-21]
```

```
team1@unix:~> dbinfo -u team1 toydb

ELOQUENCE DBINFO (C) Copyright 2008 Marxmeier Software AG (B.08.00)

Processing database: toydb

Fatal error #-4 while opening database
DBLOGON(0): Password does not match [-4]


team1@unix:~> dbinfo -u team1 -p dogbert toydb

ELOQUENCE DBINFO (C) Copyright 2008 Marxmeier Software AG (B.08.00)

Processing database: toydb

SET NAME               RECLEN CAPACITY ENTRIES
---------------- --- - ------ -------- --------
PRODUCTS         001 M     50     2553       16
CUSTOMERS        002 M    260      229        8
ORDER-MASTER     003 A      8     1456       15
INVOICES         004 D     40     1149        1
ORDERS           005 D     18     1771       15
ORDER-DETAILS    006 D     20     1771       15
PRICE-HIST       007 D     22     3542        1
```

Specifying user and password information with command-line options is not always possible, for example query3k or the sample programs like IMG01 do not accept such command-line options. From a security perspective it may also not be desirable, at least for the password (just think of the sh_history file or the „ps -ef“ command).

This is where EQ_DBUSER and EQ_DBPASSWORD come into play. They can either provide the respective user and password parameters or else reference a (carefully secured) file with that info.

```
( enter the database password to a local file without screen echo...)

team1@unix:~> ( stty -echo; read pw; stty echo; umask 077; echo $pw >my.pw )

team1@unix:~> ls -l my.pw

-rw-------  1 team1 users 8 2006-01-04 18:34 my.pw


( now using the env vars for checking write access in query3k... )

team1@unix:~> export EQ_DBUSER=team1
team1@unix:~> export EQ_DBPASSWORD=file:$PWD/my.pw

team1@unix:~> query3k

B.08.00.01 Eloquence QUERY3K  WED, JAN  4, 2006,  6:35 PM
Copyright 2004-2008 Marxmeier Software AG
Copyright 2004 Hewlett-Packard Development Company, L.P.

>base=toydb
PASSWORD = >>
MODE = >>1

>add products

PRODUCT-NO      =>>//

>exit
```

With the above customization we restricted database access to the password-protected user team1.

If we want to grant read-access to the other teams of this training, we can either create additional database users and grant them membership to the “READING” group... or add the predefined user “public” to that very group... or else grant read privileges on all datasets to the “public” group.

With the second or third approach, the other teams have to specify neither user nor password:

```
( enter the dba password to a local file without screen echo...)

team1@unix:~> ( stty -echo; read pw; stty echo; umask 077; echo $pw >dba.pw )

team1@unix:~> ls -l dba.pw

-rw-------  1 team1 users 8 2006-01-04 18:37 dba.pw


team1@unix:~> dbutil -u dba -p file:$PWD/dba.pw -

ELOQUENCE DBUTIL (C) Copyright 2008 Marxmeier Software AG (B.08.00)

database "toydb";
grant read on all to "public";
exit;


( using query3k to confirm that public user only gets read access now... )

team1@unix:~> unset EQ_DBUSER
team1@unix:~> unset EQ_DBPASSWORD

team1@unix:~> query3k

B.08.00.01 Eloquence QUERY3K  WED, JAN  4, 2006,  6:39 PM
Copyright 2004-2008 Marxmeier Software AG
Copyright 2004 Hewlett-Packard Development Company, L.P.

>base=toydb
PASSWORD = >>
MODE = >>1

>add products
ILLEGAL ACCESS

>exit
```

Notice that you can examine or customize the security settings with dbutil in interactive mode (using the forms based user interface) as well. Simply invoke dbutil without any parameters and interact with the screens displayed. For best results on HP terminals or emulators (text graphics, function keys) you should set TERM=70092 (if echo $TERM shows “hp” as your default).

While we are on the subject of database security... You want to make sure that your Eloquence database server files (config, volumes, logs) are owned by a dedicated user (your teamN in this training) and do not allow read or write access for other users. Also remember to protect the directories containing those files appropriately (remove write permissions for all other users).

Access to the database contents by regular users should only be possible through (and controlled by) your database server, so those users do not need any read/write permissions at the file-system level.

Lab 2.7 – Use the database auditing feature

In this exercise we will enable database auditing in addition to the already running forward logging. We will perform a „binary extract“ of the audit information from the forward logs to a file suitable for archiving and further processing. We will also perform a few „clear-text“ reports to inspect the contents of the extracted file.

First adjust your eloqdb.cfg file to enable auditing (you could use AuditOnly=1, if the forwardlog is not planned to be used for use with dbrecover). Before activating the config change by restarting your server instance, move the previous logfiles (not the database log volumes!) to a subdirectory log/done, including the currently active one. The server restart will start a new logfile in the original directory, so you have all logfiles with audit info in one place.

Use query3k to perform a few database updates, for example by adding a products entry, updating its price to zero, and finally deleting the entry again. This will result in audit log entries for dbput, dbupdate, and dbdelete transactions.

To extract and report the audit information, move all current forward logfiles (including the active one) to a dedicated subdirectory log/prep, and then restart the forwardlog to switch to a new logfile in the original directory. This will give you a „clean cut“ for processing the logs with fwaudit.

First use „fwaudit -o“ on the file(s) in log/prep to perform a „binary extract“ of the audit info. Using a target filename based on date and time is typically convenient. After this step, you can move the processed forward logfiles to the log/done subdirectory (or delete them, if you are not also logging for dbrecover usage).

You can examine the extracted audit log with „fwaudit -r“ now. This will produce clear-text reports. First use the –v option to get a verbose report. Then use the –I option to restrict the output to items product-no and price. Finally drill down to the dbupdate entry that changed the price to zero, using the –e option with an expression (for dbupdate and new price=0) and –v to see all items.

```
team1@unix:~> cp -p eloqdb.cfg eloqdb.old

team1@unix:~> ## customize eloqdb.cfg (see diff output below for changes)


team1@unix:~> diff eloqdb.old eloqdb.cfg

514c514
< #EnableAudit = 0
---
> EnableAudit = 1


team1@unix:~> mkdir log/prep

team1@unix:~> mv log/*.log log/done

team1@unix:~> /etc/init.d/eloq8 restart team1    # /sbin/init.d/... for hp-ux

Stopping eloqdb[team1] daemon                                       done
Starting eloqdb[team1] daemon                                       done
```

```
team1@unix:~> ls -l log

drwxr-xr-x  2 team1 users 4096 2006-01-15 21:38 done
-rw-------  1 team1 users  118 2006-01-15 21:38 fw-8-1.log
drwxr-xr-x  2 team1 users 4096 2006-01-15 21:37 prep


team1@unix:~> EQ_DBUSER=team1 EQ_DBPASSWORD=file:$PWD/my.pw  query3k

B.08.00.01 Eloquence QUERY3K  SUN, JAN 15, 2006,  9:47 PM
Copyright 2004-2008 Marxmeier Software AG
Copyright 2004 Hewlett-Packard Development Company, L.P.

>base=toydb
PASSWORD = >>
MODE = >>1

>add products
PRODUCT-NO      =>>Team1X
PRODUCT-NAME    =>>Extra Product
PRICE           =>>99
PRODUCT-LINE    =>>X
QUANTITY        =>>10
VENDOR-NAME     =>>team1 inc.

PRODUCT-NO      =>>//

>find products.product-no="Team1X"
1  ENTRIES QUALIFIED

>replace price="0"; end

>find products.product-no="Team1X"
1  ENTRIES QUALIFIED

>delete
DELETE ALL RETRIEVED ENTRIES (YES OR NO)?
>>yes

>exit


team1@unix:~> mv log/*.log log/prep

team1@unix:~> dbctl -u dba -p file:$PWD/dba.pw forwardlog restart

Forward-logging has been restarted.
Forward-log is '/home/team1/log/fw-9-1.log'.

team1@unix:~> ls -l log

drwxr-xr-x  2 team1 users 4096 2006-01-15 21:38 done
-rw-------  1 team1 users  310 2006-01-15 21:52 fw-9-1.log
drwxr-xr-x  2 team1 users 4096 2006-01-15 21:52 prep


team1@unix:~> outfile="audit-$(date +%Y%m%d-%H%M).log"

team1@unix:~> echo $outfile

audit-20060115-2159.log
```

```
team1@unix:~> fwaudit -help

ELOQUENCE FWAUDIT (C) Copyright 2008 Marxmeier Software AG (B.08.00)
usage: fwaudit [options] file [...]
options:
 -help       - show usage (this list)
 -o filename - write binary audit output to file (- for stdout)
 -c comment  - write clear-text comment into binary audit file
 -e expr     - specify filter expression
 -f filename - read filter expression from file
 -r          - print clear-text report
 -m          - print MEMO records
 -i number   - print values of the first 'number' items
 -I itemlist - print values of specified items (comma- or space-separated)
 -v          - print progress info / report details (-vv: more details)
 -x          - clear-text diagnostic output (hexdump format)


team1@unix:~> fwaudit -o $outfile -c "training" -v log/prep/*.log

ELOQUENCE FWAUDIT (C) Copyright 2008 Marxmeier Software AG (B.08.00)

processing file: log/prep/fw-8-1.log


team1@unix:~> ls -l au*

-rw-r--r--  1 team1 users 590 2006-01-15 22:02 audit-20060115-2159.log


team1@unix:~> mv log/prep/*.log log/done


team1@unix:~> fwaudit -r -v audit-20060115-2159.log

ELOQUENCE FWAUDIT (C) Copyright 2008 Marxmeier Software AG (B.08.00)

processing file: audit-20060115-2159.log

training

SIGN-ON session:10 connection time: 2006-01-15 21:48:36
 protocol{8}os{Linux}ip{127.0.0.1}user{team1}login{team1}conntime{1137358131}
 uid{1001}pid{5597}pname{query3k}

DBPUT TOYDB.PRODUCTS (#111) recno:17 session:10
 timestamp: 2006-01-15 21:48:51
  PRODUCT-NO            : "Team1X"
  PRODUCT-NAME          : "Extra Product"
  PRICE                 : 99
  PRODUCT-LINE          : "X"
  QUANTITY              : 10
  VENDOR-NAME           : "team1 inc."

DBUPDATE TOYDB.PRODUCTS (#111) recno:17 session:10
 timestamp: 2006-01-15 21:49:32
  PRODUCT-NO            : "Team1X"
  PRODUCT-NAME          : "Extra Product"
 -PRICE                 : 99
 +PRICE                 : 0
  PRODUCT-LINE          : "X"
  QUANTITY              : 10
  VENDOR-NAME           : "team1 inc."

DBDELETE TOYDB.PRODUCTS (#111) recno:17 session:10
 timestamp: 2006-01-15 21:49:50
  PRODUCT-NO            : "Team1X"
  PRODUCT-NAME          : "Extra Product"
  PRICE                 : 0
  PRODUCT-LINE          : "X"
  QUANTITY              : 10
  VENDOR-NAME           : "team1 inc."
```

```
team1@unix:~> fwaudit -r -I product-no,price audit-20060115-2159.log

ELOQUENCE FWAUDIT (C) Copyright 2008 Marxmeier Software AG (B.08.00)

training

SIGN-ON session:10 connection time: 2006-01-15 21:48:36
 protocol{8}os{Linux}ip{127.0.0.1}user{team1}login{team1}conntime{1137358131}
 uid{1001}pid{5597}pname{query3k}

DBPUT TOYDB.PRODUCTS (#111) recno:17 session:10
 timestamp: 2006-01-15 21:48:51
  PRODUCT-NO            : "Team1X"
  PRICE                 : 99

DBUPDATE TOYDB.PRODUCTS (#111) recno:17 session:10
 timestamp: 2006-01-15 21:49:32
  PRODUCT-NO            : "Team1X"
 -PRICE                 : 99
 +PRICE                 : 0

DBDELETE TOYDB.PRODUCTS (#111) recno:17 session:10
 timestamp: 2006-01-15 21:49:50
  PRODUCT-NO            : "Team1X"
  PRICE                 : 0


team1@unix:~> fwaudit -r -e "dbupdate and +price=0" -v audit-20060115-2159.log

ELOQUENCE FWAUDIT (C) Copyright 2008 Marxmeier Software AG (B.08.00)

processing file: audit-20060115-2159.log

training

SIGN-ON session:10 connection time: 2006-01-15 21:48:36
 protocol{8}os{Linux}ip{127.0.0.1}user{team1}login{team1}conntime{1137358131}
 uid{1001}pid{5597}pname{query3k}

DBUPDATE TOYDB.PRODUCTS (#111) recno:17 session:10
 timestamp: 2006-01-15 21:49:32
  PRODUCT-NO            : "Team1X"
  PRODUCT-NAME          : "Extra Product"
 -PRICE                 : 99
 +PRICE                 : 0
  PRODUCT-LINE          : "X"
  QUANTITY              : 10
  VENDOR-NAME           : "team1 inc."
```

Module 3 – Advanced Topics

Lab 3.1 – Using performance metrics

In this exercise you will enable performance metrics for your database server, run a little sample program to generate some database traffic and then examine the resulting metrics collected. You will be using both, the “live” metrics on the http status web pages that have already been enabled in lab 2.4, as well as the “long term” metrics written to StatFile and SessionStatFile text files by the database server.

For detailed documentation on the http status web pages and database server statistics also see

– http://eloquence.marxmeier.com/support/B0800/doc/http-status/index.html
– http://eloquence.marxmeier.com/support/B0800/doc/stats/index.html

To permanently enable collection of performance metrics, customize the StatFile, StatFileFlags, SessionStatFile and SessionStatMode parameters in the eloqdb.cfg file. Also verify that you still have ServiceHttp enabled (from lab 2.4). Activating config file changes normally requires a restart of the database server. However, the StatFile and SessionStatFile settings may also be controlled with dbctl commands, so we can use dbctl to avoid the server restart here.

For generating a sample database workload, we use a small Eloquence program (ordering.PROG). It uses a small PERFDB database with the same structure as the original TOYDB, so you create it with schema and dbcreate before running the ordering program.

The ordering program displays a menu with options to seed the products dataset and the customers dataset, and then populate the orders and order-details datasets with a sequence of generated orders. By accepting the default values for all prompted parameters, you will get a test run with moderate database traffic that will last for about 5 minutes elapsed time.

During this period, examine the http status web pages to review server metrics in the /perf page, as well as session metrics in the /session/x page. After the test run has finished, examine the StatFile and SessionStatFile to review the metrics collected there. You may also use two sample GNU awk scripts to reformat StatFile and SessionStatFile lines for easier reading.

Start by customizing your eloqdb.cfg file:

```
team1@unix:~> cp -p eloqdb.cfg eloqdb.old

team1@unix:~> ## customize eloqdb.cfg (see diff output below for changes)

team1@unix:~> diff eloqdb.old eloqdb.cfg
222c222
< #StatFile = /var/opt/eloquence/8.0/eloqdb.status
---
> StatFile = /home/team1/server.stats
237c237
< #StatFileFlags = ""
---
> StatFileFlags = "sa"
246c246
< #SessionStatFile = /var/opt/eloquence/8.0/eloqdb.session
---
> SessionStatFile = /home/team1/session.stats
260c260
< #SessionStatMode = 0
---
> SessionStatMode = 1
```

Verify that ServiceHttp is still enabled (from lab 2.4):

```
team1@unix:~> grep ^Service eloqdb.cfg

Service = eqdb_team1
ServiceHttp = http_team1
```

Use dbctl to review and change settings in running server:

```
team1@unix:~> dbctl statfile

statfile "DISABLED"

team1@unix:~> dbctl statfileflags

statfileflags ""

team1@unix:~> dbctl sessionstatfile

sessionstatfile "DISABLED"

team1@unix:~> dbctl sessionstatmode

sessionstatmode 0


team1@unix:~> export EQ_DBUSER=dba
team1@unix:~> export EQ_DBPASSWORD=file:$PWD/dba.pw

team1@unix:~> dbctl statfile $PWD/server.stats

statfile set to "/home/team1/server.stats".

team1@unix:~> dbctl statfileflags "sa"

statfileflags set to "sa"

team1@unix:~> dbctl sessionstatfile $PWD/session.stats

sessionstatfile set to "/home/team1/session.stats"

team1@unix:~> dbctl sessionstatmode 1

sessionstatmode set to 1


team1@unix:~> ls -l *stats

-rw-r--r-- 1 team1 users  50 2009-12-21 11:20 server.stats
-rw-r--r-- 1 team1 users 185 2009-12-21 11:19 session.stats
```

Create PERFDB database (from TOYDB schema) for ordering test program:

```
team1@unix:~> schema -T -b perfdb tmp/TOYDB.sch

ELOQUENCE SCHEMA (C) Copyright 2008 Marxmeier Software AG (B.08.00)
PAGE  1    PERFDB               ELOQUENCE SCHEMA PROCESSOR B.08.00

BEGIN DATA BASE TOYDB ;
*** (WARNING #21) Database name already specified (ignored) in line 1

Number of warnings      : 1
Item name count         : 23
Index item name count   : 0
```

```
Data set count          : 6
Collating sequence count: 0

*** Database catalog created


team1@unix:~> dbcreate perfdb

ELOQUENCE DBCREATE (C) Copyright 2008 Marxmeier Software AG (B.08.00)


team1@unix:~> unset EQ_DBUSER
team1@unix:~> unset EQ_DBPASSWORD
```

Run test program, using the suggested default values for prompted parameters:

```
team1@unix:~> ls -l tmp/ordering.PROG

-rw-r--r-- 1 team1 users 6902 2009-12-21 15:41 tmp/ordering.PROG


team1@unix:~> eloqcore $PWD/tmp/ordering

MENU
1 = seed products
2 = seed customers
3 = create orders
4 = dbinfo perfdb
0 = exit

Your choice: 1

enter database name (default = perfdb):
enter product_no min (default = 100):
enter product_no max (default = 199):
dbopen ok
dbput ok (range  100  to  199 )
dbclose ok

MENU
1 = seed products
2 = seed customers
3 = create orders
4 = dbinfo perfdb
0 = exit

Your choice: 2

enter database name (default = perfdb):
enter customer_no min (default = 1000):
enter customer_no max (default = 1999):
dbopen ok
dbput ok (range  1000  to  1999 )
dbclose ok

MENU
1 = seed products
2 = seed customers
3 = create orders
4 = dbinfo perfdb
0 = exit

Your choice: 3

enter database name (default = perfdb):
enter customer_no min (default = 1000):
enter customer_no max (default = 1999):
enter customer_no inc (default = 13):
enter product_no min (default = 100):
enter product_no max (default = 199):
enter product_no inc (default = 7):
```

```
enter order_no min (default = 10000):
enter order_no max (default = 10059):
enter line item max (default = 5):
pause time in msecs (default = 5000):
dbopen ok
```

(at this point, examine the http status web pages to monitor server and session metrics)

```
  http://yourhost:3081/perf
  http://yourhost:3081/session
  http://yourhost:3081/session/9
```

(after a few minutes, the program should complete the task and redisplay the menu)

```
dbput ok (range  10000  to  10059 )
dbclose ok

MENU
1 = seed products
2 = seed customers
3 = create orders
4 = dbinfo perfdb
0 = exit

Your choice: 0
```

Examine the StatFile and SessionStatFile logged by the eloqdb server:

```
team1@unix:~> tail server.stats

1261391220 0 0 0 0 0 0 0
1261391280 0 1 1 0 2 0 0
1261391340 0 16 16 0 32 0 0
1261391400 0 18 1 0 1 0 1
1261391460 0 25 1 0 0 0 1
1261391520 0 25 1 0 0 0 1
1261391580 0 25 1 0 0 0 1
1261391640 0 25 1 0 0 0 1
1261391700 0 6 0 0 0 0 0
1261391760 0 0 0 0 0 0 0

team1@unix:~> tail -1 session.stats    # shows a very long line...

1261391657|9|E|team1|public|2009-12-21 11:29:17|300|0|0|0|0|120|1066|0|3660|71785|
5939|3000|75813|0|240|24247|0|180|7705|0|0|0|0|180|810|0|180|413|0|0|0|0|0|0|0|0|0|
127.0.0.1:53998|uid=1003 pid=29589 pname=eloqcore /home/team1/tmp/ordering
```

Using GNU awk may be useful to format timestamp values and layout output:

```
team1@unix:~> ls -l tmp/fmt*awk

-rw-r--r-- 1 team1 users  144 2009-12-21 11:43 tmp/fmt-srv-stats.gawk
-rw-r--r-- 1 team1 users 1397 2009-12-21 11:02 tmp/fmt-ssn-stats.gawk

team1@unix:~> tail server.stats | gawk -f tmp/fmt-srv-stats.gawk

#date time timestamp util requests commits reads writes syncs clients
2009-12-21 11:28 1261391280 0 1 1 0 2 0 0
2009-12-21 11:29 1261391340 0 16 16 0 32 0 0
2009-12-21 11:30 1261391400 0 18 1 0 1 0 1
2009-12-21 11:31 1261391460 0 25 1 0 0 0 1
2009-12-21 11:32 1261391520 0 25 1 0 0 0 1
2009-12-21 11:33 1261391580 0 25 1 0 0 0 1
2009-12-21 11:34 1261391640 0 25 1 0 0 0 1
2009-12-21 11:35 1261391700 0 6 0 0 0 0 0
2009-12-21 11:36 1261391760 0 0 0 0 0 0 0
2009-12-21 11:37 1261391820 0 0 0 0 0 0 0
```

```
team1@unix:~> grep ordering session.stats | tail -1 | gawk -f tmp/fmt-ssn-stats.gawk

Timestamp       2009-12-21 11:34:17
TID             9
RecType         E
OsUser          team1
DbUser          public
ConnTime        2009-12-21 11:29:17
ConnSecs        300
IO_READ         0 0 0
DBFIND          0 120 1066  avg 8.88333
DBGET           0 3660 71785  avg 19.6134
DBGETB          5939 3000 75813  avg 25.271
DBPUT           0 240 24247  avg 101.029
DBUPDATE        0 180 7705  avg 42.8056
DBDELETE        0 0 0
DBLOCK          0 180 810  avg 4.5
DBUNLOCK        0 180 413  avg 2.29444
TXBEGIN         0 0 0
TXCOMMIT        0 0 0
TXROLLBACK      0 0 0
ClientIP        127.0.0.1:53998
AppEnv          uid=1003 pid=29589 pname=eloqcore /home/team1/tmp/ordering
```

(importing StatFile or SessionStatFile to a spreadsheet for review or charting may be helpful, too)

Note that the above example shows how to enable and review the performance metrics of the eloqdb server, but the workload caused be the ordering program run with default parameters is very moderate. You will thus not see significant CPU utilization in the server metrics, for example. As the test database is also very small compared to the default BufferCache, the performance metrics will also not show significant read or write rates to the host file system. As StatFile only records integer values, read or write rates below “1 per second” might be logged as zero.

In the above example output of server.stats we see that from 11:31 to 11:34 the average client request rate was 25 per sec and the average commit rate was 1 per sec. The average CPU utilization of the eloqdb server was below 1 percent. The average read and write rates to the operating system also were below 1 per second.

In the above example output of session.stats we see that the program was connected to the eloqdb server for 300 secs (from 11:29:17 to 11:34:17) and -for example- spent 24247 microseconds of its time doing 240 DBPUT calls, which computes to an average time of about 101 microseconds per DBPUT. These times are typically higher in real-life scenarios. In our case the database is small enough to reside in the eloqdb BufferCache, so that physical disk reads did not slow down the test run. This can be seen in the values for the IO_READ metric: we spent a total of 0 microseconds doing 0 file system reads.

You may want to re-run the ordering test program varying the database workload by using different parameters for generating orders (for example, changing order_no min and order_no max to cover a 10x or 100x bigger range and reducing pause time to 500 or 50 msecs will increase the workload by generating more orders at a faster entry rate). You may also run multiple instances of the ordering program in different sessions (using non-overlapping ranges for order_no min and order_no max).

Lab 3.2 – Using database replication

In this exercise you will create a slave server for your team's database server and implement replication.Your slave server will be running on the same system as your team's master server for this exercise; in real-life deployments, the slave will typically run on a separate system for disaster protection or workload balancing, for example. We will use the sample program from Lab 3.1 to generate some database traffic on the master server and monitor its replication to the slave.

For detailed documentation on Eloquence replication also see

- http://eloquence.marxmeier.com/support/B0800/doc/repl/index.html

To set up Eloquence replication, you first customize your eloqdb.cfg for acting as a master server. You should also verify that the forward logging option is still enabled, because the forward log files are required for implementing replication. Then create an eloqdb.cfg for your slave server, which will be running in a dedicated subdirectory of your team's home directory.

After restarting your master server to activate the config changes, you clone the master data volume files for setting up your slave server. Switch the master server to online backup mode for copying the data volume files in consistent state. Use dbvolextend with -R option to create the log volume files for the slave server.

After starting the slave server, invoke the dbrepl utility to begin replication. As dbrepl will typically be running permanently (for continuous replication), you should run it as a background process that will not exit when your interactive session ends; using nohup and I/O redirection is helpful here.

For generating a sample database workload, we use the small Eloquence program (ordering.PROG) that was already used during Lab 3.1 before. Products and customers have already been populated, so you just need to generate orders choosing a new order number range (min .. max).

While the ordering program generates database transactions, examine status and progress of the replication by using dbinfo and dbctl commands. You may also use the eloqdb status web pages and access the master or slave server with query3k, for example. For the slave server, only read access is allowed.

These are the port numbers used by the different teams for their slave servers:

```
eqdb_team1_s 4001/tcp  # team1 slave server
eqdb_team2_s 4002/tcp  # team2 slave server
eqdb_team3_s 4003/tcp  # team3 slave server
eqdb_team4_s 4004/tcp  # team4 slave server
 …

http_team1_s 4081/tcp  # team1 slave server http
http_team2_s 4082/tcp  # team2 slave server http
http_team3_s 4083/tcp  # team3 slave server http
http_team4_s 4084/tcp  # team4 slave server http
 …
```

Start by customizing your config file for the master server:

```
team1@unix:~> cp -p eloqdb.cfg eloqdb.old

team1@unix:~> ## customize eloqdb.cfg (see diff below for changes)
```

```
team1@unix:~> diff eloqdb.old eloqdb.cfg

29c29
< #Title =
---
> Title = team1 master server
296c296
< #Role = Standalone
---
> Role = Master
```

Verify that the master server is (still) configured for forward logging:

```
team1@unix:~> grep ^FwLog eloqdb.cfg

FwLog = /home/team1/log/fw-%N.log
```

Create config file for the slave server (using a dedicated subdirectory):

```
team1@unix:~> mkdir slave

team1@unix:~> cp eloqdb.cfg slave/eloqdb.cfg

team1@unix:~> ## customize slave/eloqdb.cfg (see diff below for changes)

team1@unix:~> diff eloqdb.cfg slave/eloqdb.cfg

29c29
< Title = team1 master server
---
> Title = team1 slave server
41,42c41,42
< Service = eqdb_team1
< ServiceHttp = http_team1
---
> Service = eqdb_team1_s
> ServiceHttp = http_team1_s
114c114
< LogFile = /home/team1/eloqdb.log
---
> LogFile = /home/team1/slave/eloqdb.log
222c222
< StatFile = /home/team1/server.stats
---
> StatFile = /home/team1/slave/server.stats
246c246
< SessionStatFile = /home/team1/session.stats
---
> SessionStatFile = /home/team1/slave/session.stats
296c296
< Role = Master
---
> Role = Slave
479c479
< FwLog = /home/team1/log/fw-%N.log
---
> FwLog = /home/team1/slave/log/fw-%N.log
531,532c531,532
< Root = /home/team1/db/data-01.vol
< Log02 = /home/team1/db/log-01.vol
---
> Root = /home/team1/slave/db/data-01.vol
> Log02 = /home/team1/slave/db/log-01.vol
```

Restart master server to activate its config changes:

```
team1@unix:~> /etc/init.d/eloq8 restart team1    # /sbin/init.d/... for hp-ux

Stopping eloqdb[team1] daemon                                       done
Starting eloqdb[team1] daemon                                       done
```

Clone data volumes from master to slave server (using online backup mode here):

```
team1@unix:~> dbctl -u dba -p file:$PWD/dba.pw backup start

On-line backup mode has been started.

team1@unix:~> mkdir slave/db slave/log

team1@unix:~> cp db/data-*.vol slave/db/

team1@unix:~> dbctl -u dba -p file:$PWD/dba.pw backup stop

On-line backup mode has been stopped.
```

Create fresh log volume for slave server (because we copied using online backup mode):

```
team1@unix:~> dbvoldump -c slave/eloqdb.cfg

ELOQUENCE DBVOLDUMP (C) Copyright 2008 Marxmeier Software AG (B.08.00)

ID  Type Path
1   DATA /home/team1/slave/db/data-01.vol
2   LOG  [missing]

ID  Type  Cur.Sz  Ext.Sz  Max.Sz    Free    Used
1   DATA     4.5     1.0     0.0     1.4     3.1

team1@unix:~> dbvolextend -v -c slave/eloqdb.cfg -R

ELOQUENCE DBVOLEXTEND (C) Copyright 2008 Marxmeier Software AG (B.08.00)

Reading configuration ...
Opening root volume
Restoring log volume /home/team1/slave/db/log-01.vol
Closing volume ...
done.

team1@unix:~> dbvoldump -c slave/eloqdb.cfg

ELOQUENCE DBVOLDUMP (C) Copyright 2008 Marxmeier Software AG (B.08.00)

ID  Type Path
1   DATA /home/team1/slave/db/data-01.vol
2   LOG  /home/team1/slave/db/log-01.vol

ID  Type  Cur.Sz  Ext.Sz  Max.Sz    Free    Used
1   DATA     4.5     1.0     0.0     1.4     3.1
2   LOG      2.5     1.0     0.0     2.5     0.0
```

Start the slave server (and verify successful startup):

```
team1@unix:~> /etc/init.d/eloq8 start team1_s    # /sbin/init.d/... for hp-ux

Starting eloqdb[team1_s] daemon                                       done
```

```
team1@unix:~> cat slave/eloqdb.log

** Mon Jan  4 15:29:49 2010   Flags = [*0E1]

Mon 04 15:29:49 ( 8576) D0: Server patch level PE80-0906080
Mon 04 15:29:49 ( 8576) D0: Server is configured as a replication SLAVE server
Mon 04 15:29:49 ( 8576) D0: Eloquence database server active

team1@unix:~> /etc/init.d/eloq8 status team1 team1_s    # /sbin/init.d/... for hp-ux

eloqdb[team1] process is active (pid 8458)                          running
eloqdb[team1_s] process is active (pid 8576)                        running
```

Start dbrepl process to begin replication (using dbctl and log files for status checking):

```
team1@unix:~> dbctl forwardlog status

Forward-logging is enabled.
Forward-log is '/home/team1/log/fw-14-1.log'.

team1@unix:~> dbctl -s eqdb_team1_s replication status

Server is configured as SLAVE
Replication is inactive
Last checkpoint is 14-1.1 (2010-01-04 15:26:29)
Forward-logging is enabled

team1@unix:~> dbrepl -help

usage: dbrepl [options] [slave_server_addr]
options:
 -help        - show usage (this list)
 -c cfg       - master server configuration file
 -v           - verbose, display progress
 -u name      - user name (defaults to dba)
 -p pswd      - password
 -d flags     - debug flags
 -S           - synchronize on existing log, then exit
 -G           - process current log generation, then exit
 -b bps[k|m]  - limit bandwidth to bps [kilo|mega] bits per second
 -T timestamp - process until point in time (incl.)

timestamp formats:
 YYYY-MM-DD HH:MM:SS
 MM/DD/YYYY HH:MM:SS
 DD.MM.YYYY HH:MM:SS
 note: any character may be used to separate date and time
       time part is optional (defaults to 00:00:00)


team1@unix:~> nohup dbrepl -c eloqdb.cfg -u dba -p file:$PWD/dba.pw \
>                            -v :eqdb_team1_s  </dev/null >>repl.log 2>&1 &
[1] 8850


team1@unix:~> ps -f

UID        PID  PPID  C STIME TTY          TIME CMD
team1     6966  6963  0 13:47 pts/5    00:00:00 -bash
team1     8458     1  0 15:26 pts/5    00:00:00 eloqdb32: master server
team1     8576     1  0 15:29 pts/5    00:00:00 eloqdb32: slave server
team1     8850  6966  0 15:39 pts/5    00:00:00 dbrepl -c eloqdb.cfg -u dba -p …
team1     8853  6966  0 15:39 pts/5    00:00:00 ps -f

team1@unix:~> cat repl.log

Mon 04 15:39:37 ( 8850) R1: processing forward-log file: '/home/team1/log/fw-14-1.log'
Mon 04 15:39:37 ( 8850) R1: found synchronization point with slave server
Mon 04 15:39:37 ( 8850) R1: slave server is up-to-date until 2010-01-04 15:38:02
```

```
team1@unix:~> dbctl -s eqdb_team1_s replication status

Server is configured as SLAVE
Replication is active
Last checkpoint is 14-1.12 (2010-01-04 15:38:02)
Forward-logging is enabled
Forward-log is '/home/team1/slave/log/fw-14-1.log'
```

Use PERFDB and ordering.PROG from previous lab to generate some database traffic:

```
team1@unix:~> dbinfo perfdb

ELOQUENCE DBINFO (C) Copyright 2008 Marxmeier Software AG (B.08.00)

Processing database: perfdb

SET NAME                RECLEN CAPACITY ENTRIES
---------------- --- - ------ -------- --------
PRODUCTS         001 M     30     1456      100
CUSTOMERS        002 M    260     1145     1000
ORDER-MASTER     003 A      8     1456       60
INVOICES         004 D     40        0        0
ORDERS           005 D     18     1771       60
ORDER-DETAILS    006 D     20     1771      180

team1@unix:~> eloqcore $PWD/tmp/ordering

MENU
1 = seed products
2 = seed customers
3 = create orders
4 = dbinfo perfdb
0 = exit

Your choice: 3

enter database name (default = perfdb):
enter customer_no min (default = 1000):
enter customer_no max (default = 1999):
enter customer_no inc (default = 13):
enter product_no min (default = 100):
enter product_no max (default = 199):
enter product_no inc (default = 7):
enter order_no min (default = 10000): 20000
enter order_no max (default = 10059): 20059
enter line item max (default = 5):
pause time in msecs (default = 5000):
dbopen ok
```

(at this point, monitor master and slave e.g. with http status web pages, dbctl and dbinfo)

```
  http://yourhost:3081/session
  http://yourhost:3081/db
  http://yourhost:3081/perf

  http://yourhost:4081/session
  http://yourhost:4081/db
  http://yourhost:4081/perf

  dbctl -s eqdb_team1 replication status
  dbctl -s eqdb_team1_s replication status

  dbinfo :eqdb_team1/perfdb
  dbinfo :eqdb_team1_s/perfdb
```

(after a few minutes, the program should complete the task and redisplay the menu)

```
dbput ok (range  20000  to  20059 )
dbclose ok
```

```
MENU
1 = seed products
2 = seed customers
3 = create orders
4 = dbinfo perfdb
0 = exit

Your choice: 0
```

Here is some sample dbctl and dbinfo output after the ordering test run completed:

```
team1@unix:~> dbctl -s eqdb_team1 replication status

Server is configured as MASTER
Last checkpoint is 14-1.2113 (2010-01-04 15:48:02)
Forward-logging is enabled
Forward-log is '/home/team1/log/fw-14-1.log'

team1@unix:~> dbctl -s eqdb_team1_s replication status

Server is configured as SLAVE
Replication is active
Last checkpoint is 14-1.2113 (2010-01-04 15:48:02)
Forward-logging is enabled
Forward-log is '/home/team1/slave/log/fw-14-1.log'

team1@unix:~> dbinfo :eqdb_team1/perfdb

ELOQUENCE DBINFO (C) Copyright 2008 Marxmeier Software AG (B.08.00)

Processing database: :eqdb_team1/perfdb

SET NAME                 RECLEN CAPACITY ENTRIES
---------------- --- - ------ -------- --------
PRODUCTS         001 M     30     1456      100
CUSTOMERS        002 M    260     1145     1000
ORDER-MASTER     003 A      8     1456      120
INVOICES         004 D     40        0        0
ORDERS           005 D     18     1771      120
ORDER-DETAILS    006 D     20     1771      360

team1@unix:~> dbinfo :eqdb_team1_s/perfdb

ELOQUENCE DBINFO (C) Copyright 2008 Marxmeier Software AG (B.08.00)

Processing database: :eqdb_team1_s/perfdb

SET NAME                 RECLEN CAPACITY ENTRIES
---------------- --- - ------ -------- --------
PRODUCTS         001 M     30     1456      100
CUSTOMERS        002 M    260     1145     1000
ORDER-MASTER     003 A      8     1456      120
INVOICES         004 D     40        0        0
ORDERS           005 D     18     1771      120
ORDER-DETAILS    006 D     20     1771      360
```

Notice that dbrepl automatically follows forward log switches by default:

```
team1@unix:~> ps -fp 8850

UID        PID  PPID  C STIME TTY          TIME CMD
team1     8850  6966  0 15:39 pts/5    00:00:00 dbrepl -c eloqdb.cfg -u dba -p …

team1@unix:~> dbctl -u dba -p file:$PWD/dba.pw forwardlog restart

Forward-logging has been restarted.
Forward-log is '/home/team1/log/fw-15-1.log'.
```

```
team1@unix:~> cat repl.log

Mon 04 15:39:37 ( 8850) R1: processing forward-log file: '/home/team1/log/fw-14-1.log'
Mon 04 15:39:37 ( 8850) R1: found synchronization point with slave server
Mon 04 15:39:37 ( 8850) R1: slave server is up-to-date until 2010-01-04 15:38:02
Mon 04 15:51:54 ( 8850) R1: processing forward-log file: '/home/team1/log/fw-15-1.log'
```

Note that you might need to adjust your log file housekeeping procedures or scripts to not remove forward log files from the master server before they have been replicated to the slave server by the dbrepl process. Using dbctl replication status is typically helpful in this regard.

To stop and resume replication, you may simply kill the dbrepl process. When restarting, it always obtains information from the slave server, at which point in which forward log file to resume.

Note that dbrepl also has an option -G to cause dbrepl to automatically exit when it encounters the end of the current forward log generation, as well as an option -T to cause dbrepl to automatically exit when reaching a specified date and time.

(end of labs & solutions)