Migrating to Linux with NetCOBOL and PostgreSQL

Duane Percox

Quintessential School Systems

duane@qss.com

03/28/2003

Introduction

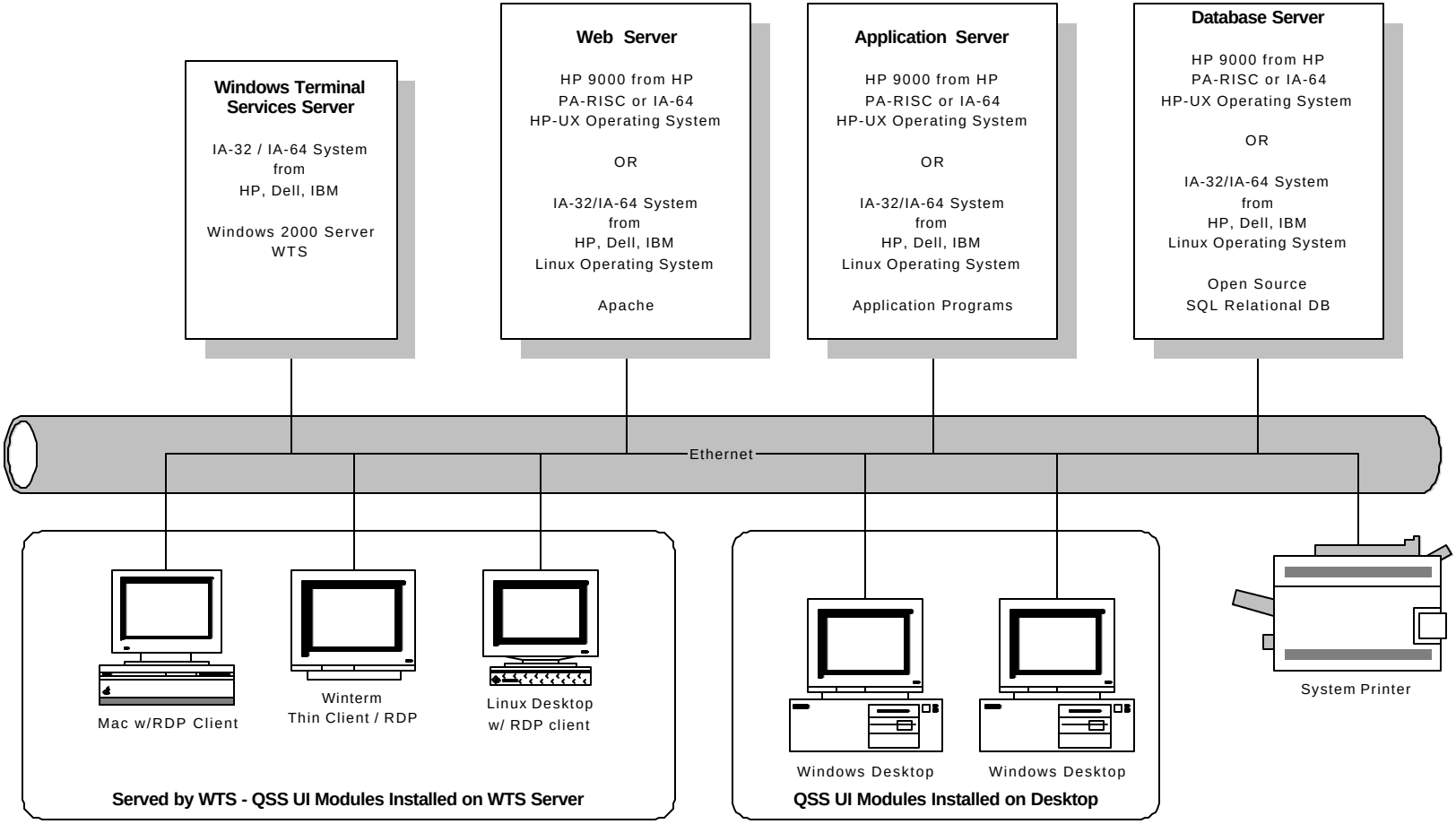
- Team: Craig Davies, Duane Percox, Jeff Woods
- A report on our on-going investigations and discoveries
- Get a PDF or PPS of this presentation:
  - qwebs.qss.com/ss3epdf.prsnt8tn.pdf
  - qwebs.qss.com/ss3epps.prsnt8tn.pps
  - qwebs.qss.com/ss3wpdf.prsnt8tn.pdf
  - qwebs.qss.com/ss3wpps.prsnt8tn.pps

Background

- Vertical market ISV
- Cost sensitive customer base: Education
- 65% turnkey installations
- MPE/iX software library mostly HP COBOL II/iX, V/Plus, TurboIMAGE, some pascal, c, spl, ksam/xl, flat files
- Migrating software in two phases: (I) UI Migration to GUI/Web and (II) server (back-end) migration to open systems (Linux/HP-UX)
- ~ 4 million lines of cobol source that we would like to preserve (app servers, reports, computational processes, data mgt processes, utilities)

QSS/OASIS and STUDENT/3000
New Platform Definition

Presentation Reference Chart #6



Database Evaluation

- Evaluate Linux open source RDBMS options and viability for QSS applications
- Evaluate creation of RDBMS versions of existing TurboIMAGE DB
- Evaluate / Determine database interface
- Establish standard data type usage
- Establish methodology for moving data
- Evaluate how to access data once it has been relocated in the RDBMS
- Evaluate the need and feasibility of creating an abstracted SQL interface for RDBMS portability

DB Evaluation Results

- Open Source RDBMS can work for QSS applications and target market
- PostgreSQL chosen as primary RDBMS
- Customer feed-back indicated a desire to have support for additional (and in a few cases commercial) RDBMS
- Decided to create an abstracted SQL interface. Our early pilot projects used a version we called DBLIB, but recently decided to name it QDBI – QSS Database Interface

PostgreSQL References

- www.postgresql.org
- PostgreSQL Essential Reference – New Riders, Stinson, 2002
- PostgreSQL Developers Handbook – SAMS, Geschwinde and Schonig, 2002
- Practical PostgreSQL – O'Reilly, Worsley & Drake, 2002
- PHP and PostgreSQL –SAMS, Geschwinde and Schonig, 2002
- PostgreSQL – A comprehensive guide to building, programming, and administering PostgreSQL databases – SAMS, Douglas and Douglas, 2003

The Case for QDBI

- Want more than an illusion of portability
  - Be able to influence performance
  - Account for DB unique SQL syntax between different DB and different versions of same DB
  - Avoid different implementations of standard interfaces
  - No changes to COBOL source
- Avoid COBOL compiler “lock-in”
- Avoid database vendor embedded SQL “lock-in”
- Allow programmers to use a COBOL friendly API
- Avoid complexity of implementing ‘standards’
- Simplify COBOL source code migration
- Simplify post migration COBOL development

RDBMS Access from COBOL

- Embedded SQL (compiler generates CALL)
  - Specific DB vendor(s) API – either because the COBOL vendor supports a limited set of DB or you use a specific DB vendor pre-compiler like Oracle Pro*COBOL
  - Compiler DB API which calls DB vendor(s) API
  - ODBC API
- CALL API directly in COBOL
  - DB specific API
  - ODBC API
  - Your own API which calls DB vendor(s) or ODBC API
- Embedded SQL with vendor neutral syntax
  - Your own pre-processor (like Pro*COBOL)
  - Generate COBOL for your choice of API

Practical Considerations

- COBOL compiler embedded SQL
  - Some don’t support embedded SQL
  - Possible syntactical differences between compilers
  - DB portability is usually qualified
  - ODBC drivers are not created equal
  - Giving up control where you might need it the most
- CALL API directly
  - Most API are designed to be called by ‘c’
  - DB vendor API are unique or they use ODBC
  - ODBC is a complex API
  - API changes occur over time

COBOL Evaluation

- Evaluate availability of Linux (open systems) COBOL compiler(s) to compile and execute QSS COBOL code with acceptable performance
- Evaluate compatibility with hp-ux COBOL and ability to maintain single source code base
- Evaluate compatibility with QSS develop. IDE (WTPS)
- Evaluate effort to move to Linux/hp-ux COBOL
- Evaluate RDBMS interface provided by COBOL compilers
- Evaluate run-time environment and issues with code generation as it relates to interfaces with QSS library routines
- Evaluate development and deployment costs

COBOL Evaluation Results

- Currently working with Fujitsu NetCOBOL
- Native compilation for Linux ia-32, no run-time costs
- Has command line compilation / link facility so don’t have to use their IDE
- Have support for hp-ux/pa-risc
- COBOL language standards give the freedom to make a change if we discover issues with this choice during the early stages of our migration

Linux Development Environment

- Dell PowerEdge 500SC server – PIII 1ghz, .5gbM, 20gbD (IDE)
- SuSe Professional 7.3
- PostgreSQL version 7.1.3 and 7.2.4
- gnu ‘c’ 2.95
- Standard Linux tools: vi, bash, ld, ar, man…
- NetCOBOL for Linux v7
- WhisperTech Programmer Studio
- Windows 2000 desktops with WRQ/Telnet and VNC/X-Windows (KDE)

Basic HP COBOL to NetCOBOL

- FTP ascii source to Linux
- '$CONTROL' → ' @OPTIONS'
- $PAGE → /
- $VERSION → no equivalent
- $COPYRIGHT → no equivalent
- PIC ... COMP → PIC ... COMP-5
  - COMP is always big-endian
  - COMP-5 is native format for platform
- WHEN-COMPILED → WHEN-COMPILED (yes!)
- CALL ... GIVING → CALL ... RETURNING
- Inline comments (find out how...)

More HP COBOL to NetCOBOL

- Copy library containing multiple entries → individual file for each referenced COPY
- <> → NOT =
- VALUE %nn (octal) → VALUE X”nn” (hex)
- Concatenation in MOVE
  - MOVE “MY TEXT” && X”00” TO WS-TEXT

More HP COBOL to NetCOBOL

- Open systems are case sensitive and this impacts COBOL file names, COPY names, and CALL “....” names.
- SELECT MY-FILE ASSIGN TO “XFILE”
  - Accesses the file XFILE
  - No file change at run time because there are no file equations
- SELECT MY-FILE ASSIGN TO XFILE
  - At run time the variable XFILE is used to determine the location of MY-FILE.
  - This can be an internal variable in working storage or an exported variable set in the shell: export XFILE=./xfile.mulder

Makefile for Simple Programs

```
prime : prime.o
 cobol -dy  -o prime  prime.o

prime.o : prime.cob
 cobol -dy -M -c -o prime -WC,"SRF(FIX)“ prime.cob

primebe : primebe.o
   cobol -dy -o primebe primebe.o

primebe.o : primebe.cob
   cobol -dy -M -c -o primebe -WC,"SRF(FIX)" primebe.cob
```

Makefile for Building Shared Lib

```
libqss.so : libqss.a
	ld -m elf_i386 -shared -o libqss.so --whole-archive libqss.a
libqss.a : module1.o module2.o proctime.o modulec.o
	ar cr libqss.a module1.o module2.o proctime.o modulec.o
module1.o : module1.cob
	cobol -shared -c -o module1.o module1.cob
module2.o : module2.cob
	cobol -shared -c -o module2.o module2.cob
proctime.o : proctime.c
	gcc -fPIC -c -o proctime.o proctime.c
modulec.o : modulec.c
	gcc -fPIC -c -o modulec.o modulec.c
modtst : libqss.so modtst.cob
	cobol -dy -M -o modtst -L. -lqss modtst.cob
```

TurboIMAGE vs PostgreSQL

- TurboImage access is direct with global structures used to control access
- RDBMS is through a connection to a DB engine in a client/server fashion
- Flexibility in DB access topology is provided natively in RDBMS and must be engineered for TurboIMAGE

PostgreSQL Basics

- Connection from client to db is transparent regardless if same system (shared memory) or different system (tcp/ip). X-system can use SSL for secure transmission.
- Server engine is called ‘postmaster’
- Separate process created for each connection. Good performance on unix style o/s since was written for unix
- DB are organized into clusters. Each instance of postmaster provides access to all DB within the cluster
- Normal installation root dir is /usr/local/pgsql
- Default cluster root is /usr/local/pgsql/data ($PGDATA)
- Databases are stored in $PGDATA/base with a separate directory for each DB and individual files for each table
- The directory structure and files (DB) owned by postgres user and not viewable/accessible by ordinary users

PostgreSQL Versioning

- Versions are formatted as x.y.z ; x is major, y is minor, z is fix level
- Current major version is 7 and base versions found since 2001 are 7.1, 7.2 and 7.3
- Usually require a DB conversion between base versions as the DB structure could change
- Pre-built PostgreSQL found on distributions will be installed in /usr/local/pgsql root with directories like bin, lib, include, data for specific components and database files
- QSS initially installed 7.1.3 and just recently converted to 7.2.4
- Conversion to 7.2.4 was painless and no software changes were required

Managing Multiple Versions

- We wanted 7.1.3 and 7.2.4 installation and data available for testing and review
- Multiple versions of PostgreSQL can be installed and running at the same time. At run time your $PGDATA, $PATH, and $LD_LIBRARY_PATH can be used to define the version
- We de-installed the original 7.1.3 version which was installed in the default directory and built from the source cvs tree 7.1.3 and 7.2.4 with the root directory set as /usr/local/pgsql/7_1_3 and /usr/local/pgsql/7_2_4
- We defined the db clusters as ~postgres/data-7_1_3 and ~postgres/data-7_2_4
- When building PostgreSQL from source you can set these locations as defaults which is then used by the postmaster and utility programs

Accessing Specific Version

- We setup soft links which would refer to the current and previous versions of PostgreSQL
- /usr/local/pgsql/current → /usr/local/pgsql/7_2_4
- /usr/local/pgsql/prevsious → usr/local/pgsql/7_1_3
- By setting $PGDATA
  - export PGDATA=~postgres/current
  - export PGDATA=~postgres/previous
- Use $PGLOC for root of version
  - export PGLOC=/usr/local/pgsql/current
  - export PGLOC=/usr/local/pgsql/previous

Making QDBI

```
libqdbi.so : libqdbi.a
    ld -m elf_i386 -shared -o libqdbi.so --whole-archive libqdbi.a

libqdbi.a : qdbi.o
    ar cr libqdbi.a qdbi.o

qdbi.o : qdbi.c
    gcc -fPIC -I${PGLOC}/include -c -o qdbi.o qdbi.c
```

Making COBOL that Calls QDBI

```
fastpg : fastpg.cob
    cobol -dy -M -o fastpg -L. -lqdbi -L ${PGLOC}/lib -lpq fastpg.cob

# Make sure the following vars are set:
#
# COBCOPY=./
#     this defines the directory where the 'copy' modules are found
#
# COB_LIBSUFFIX
#     if NOT defined the copy members will be found using .cbl/.cob
#     set if you want to use something other than .cbl/.cob (cpy or CPY)
```

PostgreSQL Client Access

- psql – character mode DB utility
- pg_access – graphical (x-windows) DB utility
- pgadmin –graphical (win 9x/nt) DB utility

PostgreSQL Programmatic Access

- libpq – ‘c’ library for access to DB
- libpq++ - ‘c++’ library for access to DB
- libpgeasy – higher level/simpler access ‘c’ library (resolves to libpq)
- libpsqlodbc – unix style ODBC driver (resolves to libpq)
- jdbc – java DB driver

Notes on Programmatic Access

- PostgreSQL doesn't use the prepare/execute model
- You can return the entire result set to the client or you can define a cursor and fetch rows in blocks from the server. Each fetch will return metadata so be careful how much you fetch
- With libpq you don't bind the columns to a memory address (ODBC does this) but you access through libpq functions each column of the row as a null terminated ascii string
- You can use binary cursors which return raw data but you have to know how to decode (be very, very careful...)

PostgreSQL Examples

- Creating a DB
- Sample psql Session
- Sample pgAdmin Session
- Sample Data Import

Create a DB

linux.r1w - Reflection for HP

File Edit Connection Setup Macro Window Help

```
craigd@linuxdev:~> createdb testdb
CREATE DATABASE
craigd@linuxdev:~> psql testdb
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

testdb=# _
```

Upd-Next /Visual | Roll Up Screen | Findup String | Find String | Backward One Page | Forward One Page | Do ===> Listredo | Exit

36, 10 | VT220-7 -- linuxdev via TELNET | Num

Define a Sample Table

linux.r1w - Reflection for HP

File Edit Connection Setup Macro Window Help

```
testdb=# \i ./sql/createtesttable.sql
DROP
CREATE
testdb=# \d qss_test_table
            Table "qss_test_table"
    Attribute     |      Type      | Modifier
-----------------+----------------+----------
 char_data_2      | character(2)   |
 char_data_30     | character(30)  |
 date_data        | date           |
 number_data_1    | numeric(12,2)  |
 number_data_2    | numeric(8,0)   |
 number_data_3    | numeric(10,4)  |
 number_data_neg  | numeric(8,2)   |
 number_data_4    | numeric(10,0)  |

testdb=#
```

Upd-Next /Visual | Roll Up Screen | Findup String | Find String | Backward One Page | Forward One Page | Do ===> Listredo | Exit

42, 10 | VT220-7 -- linuxdev via TELNET | Num

Sample Table Script

Linux - Programmer Studio - [//Linuxdev/home/craigd/sql/createtesttable.sql*]

File Edit View Insert Format Project Build Debug Vcs Tools Window Help Comments?

createtesttable.sql*

```sql
/*
   Name: TESTDB SQL Schema.
   Date: September 2002.

History:

   09/23/02 CD      Initial Schema Creation
*/

/* Set up the qss_test_table Table */
DROP   TABLE   qss_test_table;
CREATE TABLE   qss_test_table(
               char_data_2          char(2),
               char_data_30         char(30),
               date_data            date,
               number_data_1        numeric(12,2),
               number_data_2        numeric(8),
               number_data_3        numeric(10,4),
               number_data_neg      numeric(8,2),
               number_data_4        numeric(10,0));
```

Ready Ovr Tab Read Ln 5, Col 9

Load Test Data

linux.r1w - Reflection for HP

File Edit Connection Setup Macro Window Help

```
testdb=# \i ./sql/loadtable.sql
INSERT 1392283 1
testdb=# select * from qss_test_table;
 char_data_2 |          char_data_30          | date_data  | number_data_1 | num
ber_data_2 | number_data_3 | number_data_neg | number_data_4
-------------+--------------------------------+------------+---------------+----
-----------+---------------+-----------------+---------------
 AB          | A THRITY CHARACTER FIELD 1     | 2002-03-14 |    1234567.12 |
    654321 |   132445.1234 |         -123.31 |       3333333
(1 row)

testdb=#
```

Upd-Next /Visual | Roll Up Screen | Findup String | Find String | Backward One Page | Forward One Page | Do ===> Listredo | Exit

37, 10 | VT220-7 -- linuxdev via TELNET | Num

Sample Load Script

Linux - Programmer Studio - [//Linuxdev/home/craigd/sql/loadtable.sql*]

File Edit View Insert Format Project Build Debug Vcs Tools Window Help Comments?

loadtable.sql*

```
/* Name: TESTDB SQL Schema.
   Date: September 2002.

History:

   09/23/02 CD      Load data into Table
*/

 INSERT INTO  qss_test_table(char_data_2,
                             char_data_30,
                             date_data,
                             number_data_1,
                             number_data_2,
                             number_data_3,
                             number_data_neg,
                             number_data_4 )
                VALUES ('AB',
                        'A THRITY CHARACTER FIELD 1',
                        '03/14/2002',
                        1234567.12,
                        654321,
                        132445.1234,
                        -123.31,
                        3333333);
```

Ovr Tab Read Ln 1, Col 3

Sample psql Session

linux.r1w - Reflection for HP

File Edit Connection Setup Macro Window Help

```
Welcome to SuSE Linux 7.3 (i386) - Kernel 2.4.10-4GB (0).

linuxdev login: craigd
Password:
Last login: Fri Jul 19 09:34:37 from sm-dhcp-92.qss.com
Have a lot of fun...

"I haven't lost my mind -- it's backed up on tape somewhere."

craigd@linuxdev:~> psql fasset
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

fasset=#
```

Upd-Next /Visual | Roll Up Screen | Findup String | Find String | Backward One Page | Forward One Page | Do ===> Listredo | Exit

19, 10 | VT220-7 -- linuxdev via TELNET | Num

More Sample psql Session

linux.r1w - Reflection for HP

File Edit Connection Setup Macro Window Help

```
fasset-# \d qss_fixed_asset
          Table "qss_fixed_asset"
    Attribute     |      Type       | Modifier
-----------------+-----------------+----------
 di_no            | character(2)    |
 id_no            | character(10)   |
 ca_no            | character(2)    |
 ty_no            | character(4)    |
 name             | character(30)   |
 serial           | character(20)   |
 tag              | character(10)   |
 si_no            | character(4)    |
 department       | character(8)    |
 bldg             | character(8)    |
 floor            | character(8)    |
 room             | character(8)    |
 loc_descriptor   | character(30)   |
 gl_fund          | character(8)    |
 gl_subfund       | character(8)    |
 gl_object        | character(8)    |
 project_loc      | character(8)    |
 project_sta      | character(8)    |
 project_fed      | character(8)    |
 unit_cost        | numeric(12,2)   |
```

Upd-Next /Visual | Roll Up Screen | Findup String | Find String | Backward One Page | Forward One Page | Do ===> Listredo | Exit

71, 12 | VT220-7 -- linuxdev via TELNET | Num

And More psql …

linux.r1w - Reflection for HP

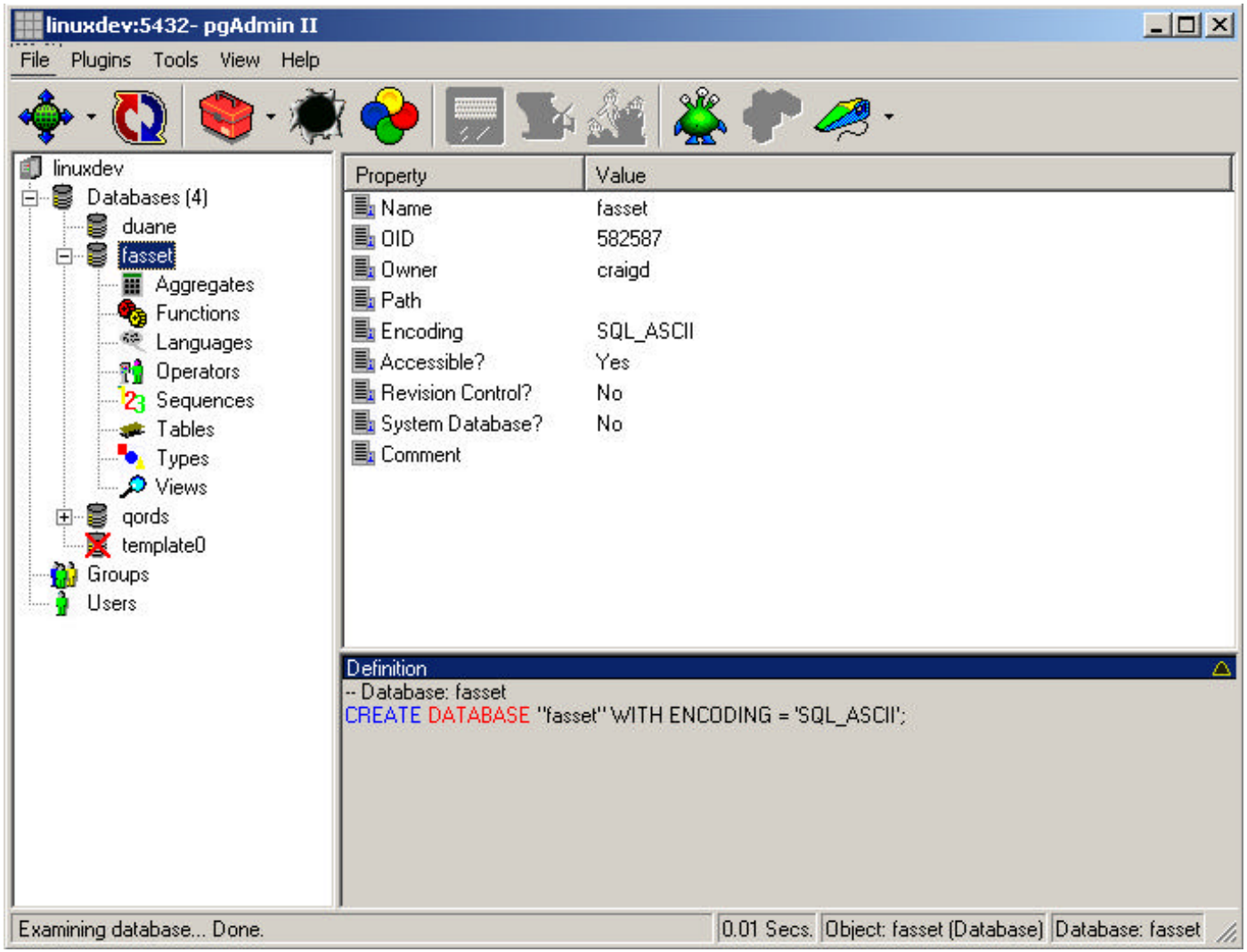
File Edit Connection Setup Macro Window Help

```
fasset=# select * from qss_fixed_asset;
 di_no |   id_no    | ca_no | ty_no |              name              |        se
-------+------------+-------+-------+--------------------------------+----------
 03    | 0000000852 | 01    | 0002  | COUNTING BEADS                 |
 03    | 0000000853 | 01    | 0001  | 2 STUDENT DESK OPEN FRONT      |
 03    | 0000000854 | 01    | 0001  | FILE LETTER 2 DRAWER           |
 03    | 0000000855 | 01    | 0002  | GLOBE 12"                      |
 03    | 0000000856 | 01    | 0002  | MAP ROLLER 60" OVER            |
 03    | 0000000857 | 01    | 0002  | RACK CHART PRIMARY METAL       |
 03    | 0000000858 | 01    | 0001  | ROUND TABLE WOOD               |
 03    | 0000000859 | 01    | 0003  | SCREEN PROJECTION 4 - 6"       |
 03    | 0000008510 | 01    | 0001  | RECTANGLE TABLE                |
 03    | 0000008511 | 01    | 0001  | KIDNEY SHAPED TABLE 60-72"     |
 03    | 0000008512 | 01    | 0001  | TEACHERS CHAIR ON ROLLERS      |
 03    | 0000008513 | 01    | 0001  | TEACHERS PLASTIC CHAIR         |
 03    | 0000008514 | 01    | 0001  | TEACHERS DESK WOOK             |
 03    | 0000008515 | 01    | 0001  | CHAIRS                         |
 03    | 0000000891 | 01    | 0003  | UTILITY CART                   |
 03    | 0000008516 | 01    | 0001  | TEACHERS CHAIR PLASTIC         |
 03    | 9200000001 | 62    | 6204  | ADMINISTRATION BUILDING        |
 03    | 6800000001 | 62    | 6204  | SCHENDEL SCHOOL CAFETERIA      |
 03    | 6800000002 | 62    | 6205  | RELOCATABLE DISTRICT FREEZER   |
 03    | 6800000003 | 62    | 6204  | STORAGE                        |
 03    | 5200000001 | 62    | 6204  | SCHENDEL OFFICE CLASSROOM      |
```

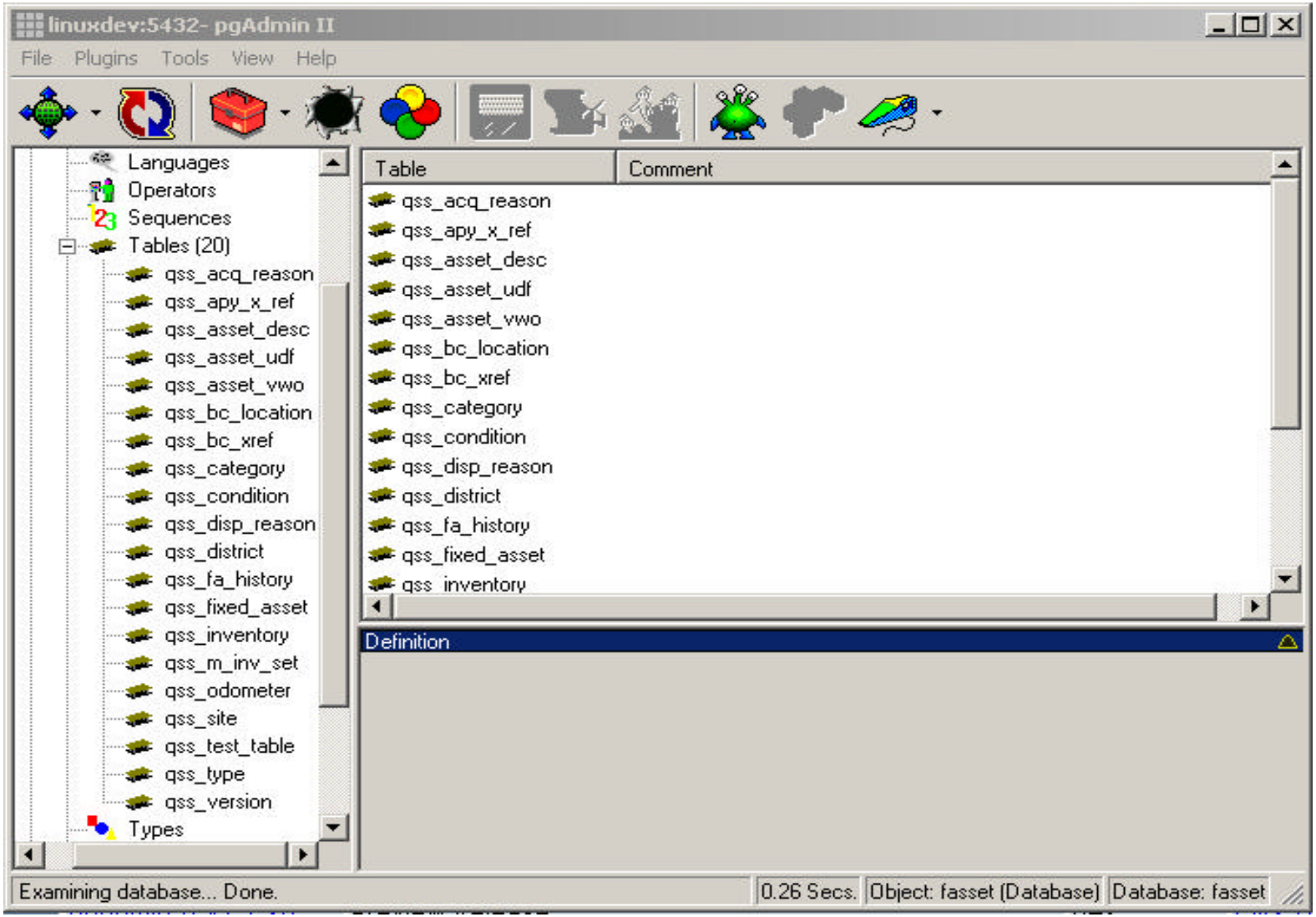
Upd-Next /Visual | Roll Up Screen | Findup String | Find String | Backward One Page | Forward One Page | Do ===> Listredo | Exit

98, 12 | VT220-7 -- linuxdev via TELNET | Num | Stop

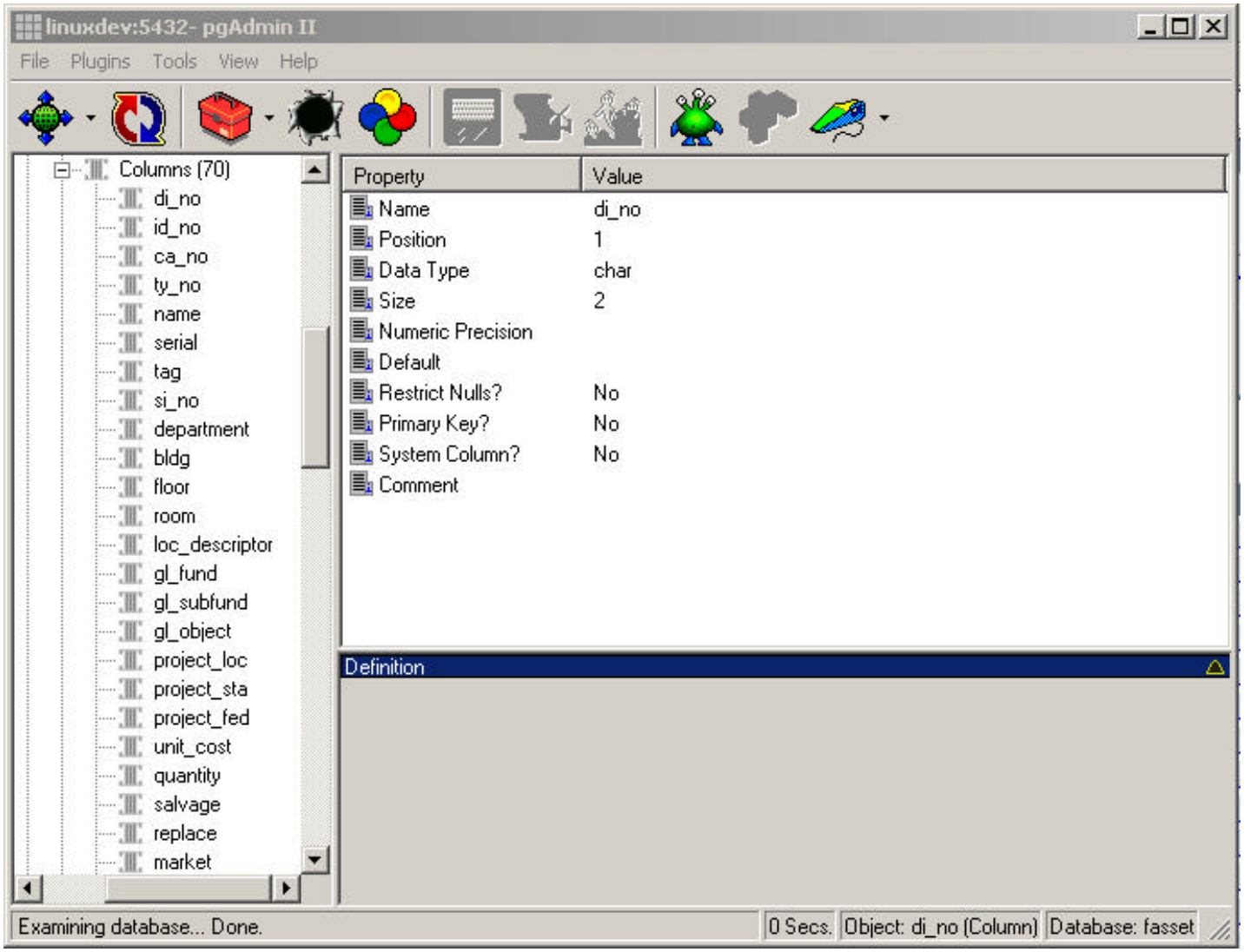
Sample pgAdmin Session



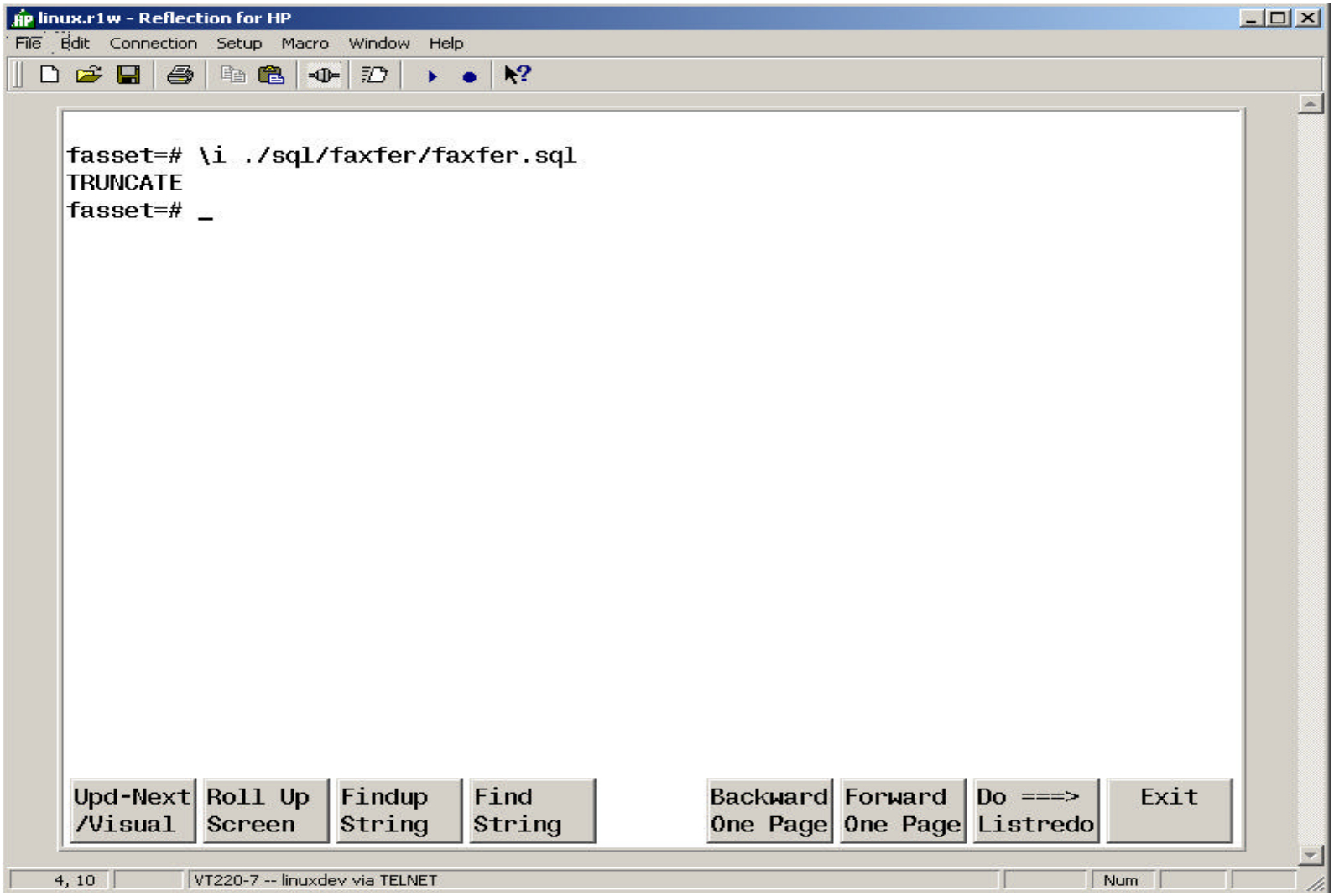
More Sample pgAdmin



And More pgAdmin



Sample Data Import



Data Import Script

```
TRUNCATE TABLE qss_fixed_asset;
COPY qss_fixed_asset FROM stdin USING DELIMITERS '~';
03~0000000851~01~0001~CHAIRS                          ~                    ~000000085
03~0000000852~01~0002~COUNTING BEADS                  ~                    ~000000085
03~0000000853~01~0001~2 STUDENT DESK OPEN FRONT       ~                    ~000000085
03~0000000854~01~0001~FILE LETTER 2 DRAWER            ~                    ~000000085
03~0000000855~01~0002~GLOBE 12"                       ~                    ~000000085
03~0000000856~01~0002~MAP ROLLER 60" OVER             ~                    ~000000085
03~0000000857~01~0002~RACK CHART PRIMARY METAL        ~                    ~000000085
03~0000000858~01~0001~ROUND TABLE WOOD                ~                    ~000000085
03~0000000859~01~0003~SCREEN PROJECTION 4 - 6"        ~                    ~000000085
03~0000008510~01~0001~RECTANGLE TABLE                 ~                    ~000000851
03~0000008511~01~0001~KIDNEY SHAPED TABLE 60-72"      ~                    ~000000851
03~0000008512~01~0001~TEACHERS CHAIR ON ROLLERS       ~                    ~000000851
03~0000008513~01~0001~TEACHERS PLASTIC CHAIR          ~                    ~000000851
03~0000008514~01~0001~TEACHERS DESK WOOK              ~                    ~000000851
03~0000008515~01~0001~CHAIRS                          ~                    ~000000851
03~0000000891~01~0003~UTILITY CART                    ~                    ~000000089
03~0000008516~01~0001~TEACHERS CHAIR PLASTIC          ~                    ~000000851
03~9200000001~62~6204~ADMINISTRATION BUILDING         ~                    ~920000000
03~6800000001~62~6204~SCHENDEL SCHOOL CAFETERIA       ~                    ~680000000
03~6800000002~62~6205~RELOCATABLE DISTRICT FREEZER    ~                    ~680000000
03~6800000003~62~6204~STORAGE                         ~                    ~680000000
03~5200000001~62~6204~SCHENDEL OFFICE CLASSROOM       ~                    ~520000000
03~6200000001~62~6204~5 CLASSROOMS                    ~                    ~620000000
03~5200000002~62~6204~CLASSROOMS                      ~                    ~520000000
```

Pilot Migration Project

- Asset Database
- Detail Set (FIXED-ASSET) with 70 fields
- 2-character path (DI-NO) and a 12-char path (ASSET-ID)
- Test programs to mirror find/get of large sets of records

TurboImage Set Definition

hp3k.r1w - Reflection for HP

File Edit Connection Setup Macro Window Help

```
FIXED-ASSET,DETAIL
   ITEMS:
      KEY,                X12          <<SEARCH ITEM>>
      DI-NO,              Z2           <<SEARCH ITEM>>
      CA-NO,              Z2
      TY-NO,              Z4
      NAME,               X30
      SERIAL,             X20
      TAG,                X10
      SI-NO,              Z4
      DEPARTMENT,         X8
      BLDG,               X8
      FLOOR,              X8
      ROOM,               X8
      LOC-DESCRIPTOR,     X30
      GL-FUND,            X8
      GL-SUBFUND,         X8
      GL-OBJECT,          X8
      PROJECT-LOC,        X8
      PROJECT-STA,        X8
      PROJECT-FED,        X8
      UNIT-COST,          P12
      QUANTITY,           P12
      SALVAGE,            P12
```

Upd-Next /Visual | Roll Up Screen | Findup String | Find String | Backward One Page | Forward One Page | Do ===> Listredo | Exit

Create, open, and save settings, start new sessions, transfer files, print, log, and quit Reflection

PostgreSQL Table Definition

Linux - Programmer Studio - [//Linuxdev/home/craigd/sql/fasset.sql*]

File Edit View Insert Format Project Build Debug Vcs Tools Window Help Comments?

ix_qss_version_dx_ca_no

fasset.psql | fasset.sql*

```
CREATE TABLE    qss_fixed_asset(
                di_no                   char(2),
                id_no                   char(10),
                ca_no                   char(2),
                ty_no                   char(4),
                name                    char(30),
                serial                  char(20),
                tag                     char(10),
                si_no                   char(4),
                department              char(8),
                bldg                    char(8),
                floor                   char(8),
                room                    char(8),
                loc_descriptor          char(30),
                gl_fund                 char(8),
                gl_subfund              char(8),
                gl_object               char(8),
                project_loc             char(8),
                project_sta             char(8),
                project_fed             char(8),
                unit_cost               numeric(12,2),
                quantity                numeric(11),
...
CREATE INDEX    ix_qss_fixed_asset_dx_di_no
       ON       qss_fixed_asset  (di_no);
CREATE INDEX    ix_qss_fixed_asset_dx_id_no
       ON       qss_fixed_asset  (di_no,id_no);
```

Ready | Ovr | Tab | Read | Ln 241, Col 5

TI Record Layout

QSSCRAIG - Programmer Studio - [//PGMR.QSSDEV/QSSDEV/COPYLIB/COPYLIB/FAFAREC]

File Edit View Insert Format Project Build Debug Vcs Tools Window Help Comments?

FAFAREC

```
*************************************************************
* * 11/14/97 SHA      Version H.                          ***
* * Database FASSET:  data set: FIXED-ASSET               ***
* *                                                       ***
* * 09/02/02 DAP  Define project fields as a table       ***
*************************************************************
     03  FA-KEY.
         05  FA-REF-DI                  PIC 99.
         05  FA-ID-NO                   PIC 9(10).
     03  FA-DI-NO                       PIC 99.
     03  FA-CA-NO                       PIC 99.
     03  FA-TY-NO                       PIC 9(4).
     03  FA-DESC                        PIC X(30).
     03  FA-SERIAL                      PIC X(20).
     03  FA-TAG                         PIC X(10).
     03  FA-SI-NO                       PIC 9(4).
     03  FA-DEPARTMENT                  PIC X(8).
     03  FA-BLDG                        PIC X(8).
     03  FA-FLOOR                       PIC X(8).
     03  FA-ROOM                        PIC X(8).
     03  FA-LOC-DESCRIPTOR              PIC X(30).
     03  FA-GL-FUND                     PIC X(8).
     03  FA-GL-SUBFUND                  PIC X(8).
     03  FA-GL-OBJECT                   PIC X(8).
     03  FA-PROJECT-VALUES.
         05  FA-PROJECT-VALUE           PIC X(8)
                                        OCCURS 3 TIMES.
```

Ready Ovr Tab Read Ln 1, Col 1

TI Record Layout cont.

QSSCRAIG - Programmer Studio - [//PGMR.QSSDEV/QSSDEV/COPYLIB/COPYLIB/FAFAREC]

File Edit View Insert Format Project Build Debug Vcs Tools Window Help Comments?

FAFAREC

```
          03  FA-PROJ-VALUES                REDEFINES
                                             FA-PROJECT-VALUES.
              05   FA-PROJECT-LOC           PIC X(8).
              05   FA-PROJECT-STA           PIC X(8).
              05   FA-PROJECT-FED           PIC X(8).
          03  FA-UNIT-COST                  PIC S9(9)V99 COMP-3.
          03  FA-QUANTITY                   PIC S9(11) COMP-3.
          03  FA-SALVAGE                    PIC S9(9)V99 COMP-3.
          03  FA-REPLACE                    PIC S9(9)V99 COMP-3.
          03  FA-MARKET                     PIC S9(9)V99 COMP-3.
          03  FA-MODEL                      PIC X(10).
          03  FA-COLOR                      PIC X(8).
          03  FA-LICENSE                    PIC X(10).
          03  FA-REG-EXPIRE-DATE            PIC 9(8) COMP.
          03  FA-INS-EXPIRE-DATE            PIC 9(8) COMP.
          03  FA-VEHICLE-NO                 PIC X(10).
          03  FA-MANUFACTURER               PIC X(20).
          03  FA-MANU-DATE                  PIC 9(8) COMP.
          03  FA-MANU-PART-NO               PIC X(20).
          03  FA-INSURANCE-CODE             PIC X(4).
          03  FA-ACQ-REASON                 PIC XX.
          03  FA-DATE-ACQ                   PIC 9(8) COMP.
*         03  FA-ACQ-R                      REDEFINES FA-DATE-ACQ
*             05   FA-YEAR-ACQ              PIC 9999.
*             05   FA-MNTH-ACQ              PIC 99.
*             05   FA-DAY-ACQ               PIC 99.
          03  FA-VENDOR-NO                  PIC 9(6).
```

Ready Ovr Tab Read Ln 1, Col 1

Pg Record Layout

Linux - Programmer Studio - [//Linuxdev/home/craigd/copylib/farecord.cpy]

File Edit View Insert Format Project Build Debug Vcs Tools Window Help Comments?

farecord.cpy

```
01  QSS-FIXED-ASSET-BUFF.
    03 FA-DI-NO                    PIC X(02).
    03 FA-ID-NO                    PIC X(10).
    03 FA-CA-NO                    PIC X(02).
    03 FA-TY-NO                    PIC X(04).
    03 FA-DESC                     PIC X(30).
    03 FA-SERIAL                   PIC X(20).
    03 FA-TAG                      PIC X(10).
    03 FA-SI-NO                    PIC X(04).
    03 FA-DEPARTMENT               PIC X(8).
    03 FA-BLDG                     PIC X(8).
    03 FA-FLOOR                    PIC X(8).
    03 FA-ROOM                     PIC X(8).
    03 FA-LOC-DESCRIPTOR           PIC X(30).
    03 FA-GL-FUND                  PIC X(8).
    03 FA-GL-SUBFUND               PIC X(8).
    03 FA-GL-OBJECT                PIC X(8).
    03 FA-PROJECT-LOC              PIC X(8).
    03 FA-PROJECT-STA              PIC X(8).
    03 FA-PROJECT-FED              PIC X(8).
    03 FA-UNIT-COST                PIC S9(10)V99.
    03 FA-QUANTITY                 PIC S9(11).
    03 FA-SALVAGE                  PIC S9(10)V99.
    03 FA-REPLACE                  PIC S9(10)V99.
    03 FA-MARKET                   PIC S9(10)V99.
    03 FA-MODEL                    PIC X(10).
    03 FA-COLOR                    PIC X(8).
```

Ready Ovr Tab Read Ln 178, Col 34

Pg Record Layout cont.

Linux - Programmer Studio - [//Linuxdev/home/craigd/copylib/farecord.cpy]

File Edit View Insert Format Project Build Debug Vcs Tools Window Help Comments?

farecord.cpy

```
      03 FA-LICENSE                    PIC X(10).
      03 FA-REG-EXPIRE-DATE            PIC X(8).
      03 FA-INS-EXPIRE-DATE            PIC X(8).
      03 FA-VEHICLE-NO                 PIC X(10).
      03 FA-MANUFACTURER               PIC X(20).
      03 FA-MANU-DATE                  PIC X(8).
      03 FA-MANU-PART-NO               PIC X(20).
      03 FA-INSURANCE-CODE             PIC X(4).
      03 FA-ACQ-REASON                 PIC XX.
      03 FA-DATE-ACQ                   PIC 9(8).
      03 FA-VENDOR-NO                  PIC 9(6).
      03 FA-VENDOR-NAME                PIC X(30).
      03 FA-WARRANT-NO                 PIC 9(8).
      03 FA-WARRANT-DATE               PIC 9(8).
      03 FA-PO-NO                      PIC X(6).
      03 FA-INVOICE-NO                 PIC X(12).
      03 FA-ACCTCLASS                  PIC X(40).
      03 FA-WARRANTY-DATE              PIC 9(8).
      03 FA-WARRANTY-LEN.
         05   FA-WARRANTY-YEARS        PIC 9(4).
         05   FA-WARRANTY-MONTHS       PIC 9(4).
         05   FA-WARRANTY-DAYS         PIC 9(4).
      03 FA-CONDITION                  PIC X(02).
      03 FA-DISP-REASON                PIC X(02).
      03 FA-DISP-DATE                  PIC 9(8).
      03 FA-DATE-CHGD                  PIC 9(8).
      03 FA-STATUS                     PIC XX.
```

Ready Ovr Tab Read Ln 178, Col 34

Sample Code (1 of 3)

QSSCRAIG - Programmer Studio - [//PGMR.QSSDEV/QSSDEV/CBLXCCD/FAS071]

File Edit View Insert Format Project Build Debug Vcs Tools Window Help Comments?

100-OPEN-DATABASES

FAS071 FAS070

```
      CALL "DBOPEN" USING                 FA-BNAME,
                                          FA-PWD,
                                          FA-MODE,
                                          DB-STATUS

      IF DB-ERR NOT = ZERO
         CALL "DBEXPLAIN" USING           DB-STATUS
         STOP RUN
      END-IF

      CALL "qdb_connect" USING            DB-FASSET,
                                          QDB-STATUS

      IF NOT DB-GOOD-STATUS THEN
         DISPLAY "*****************************************"
         DISPLAY "* DB Open Error    : " DB-FASSET-NAME
         DISPLAY "* DB ID            : " DB-FA-ID
         DISPLAY "* Return Error     : " DB-RETURN-STATUS
         DISPLAY "* qdb_init STATUS  : " DB-OTHER-STATUS
         DISPLAY "*"
         DISPLAY "* aBNoRmAl tErMiNAtIOn!"
         DISPLAY "*****************************************"
         STOP RUN
      END-IF
      .
```

Ovr Tab Read Ln 527, Col 45

Sample Code (2 of 3)

QSSCRAIG - Programmer Studio - [//PGMR.QSSDEV/QSSDEV/CBLXCCD/FAS071]

File Edit View Insert Format Project Build Debug Vcs Tools Window Help Comments?

EXTRACT-DATA

FAS071 FAS070

```
      CALL "DBFIND" USING                 FA-BNAME,
                                          FA-FILE,
                                          DB-MODE5,
                                          DB-STATUS,
                                          FA-ITEM,
                                          FX-RECORD
      IF NOT DB-SUCCESS
         GO TO ED-EXIT
      END-IF

      INITIALIZE SQL-PASS-INFO
      MOVE FAL-DI-NO                      TO SQL-FIELD(1)
      MOVE SQL-EQ                         TO SQL-OPERATOR-1(1)
      MOVE DIST-NO                        TO SQL-VALUE(1)
      MOVE SPACES                         TO SQL-OPERATOR-2(1)

      CALL "qdb_select" USING DB-FASSET,
                              QDB-STATUS,
                              QSS-FIXED-ASSET-TABLE,
                              QSS-FIXED-ASSET-LIST,
                              QSS-FIXED-ASSET-META-DATA,
                              SQL-CRITERIA

      IF NOT DB-GOOD-STATUS  THEN
         DISPLAY "No FA Records Found!"
         GO TO ED-EXIT
      END-IF
```

Ovr Tab Read Ln 583, Col 12

Sample Code (3 of 3)

QSSCRAIG - Programmer Studio - [//PGMR.QSSDEV/QSSDEV/CBLXCCD/FAS071]

File Edit View Insert Format Project Build Debug Vcs Tools Window Help Comments?

EXTRACT-DATA

FAS071 FAS070

```
      CALL "DBGET" USING                  FA-BNAME,
                                          FA-FILE,
                                          DB-MODE,
                                          DB-STATUS,
                                          DB-ALL-ITEMS,
                                          FX-RECORD,
                                          FX-RECORD.
      IF NOT DB-SUCCESS
         IF DB-END-CHAIN
            GO TO ED-EXIT
         ELSE
            PERFORM FA-ERROR
      END-IF

      CALL "qdb_getrecord" USING          DB-FASSET,
                                          QDB-STATUS,
                                          NEXT-RECORD,
                                          QSS-FIXED-ASSET-TABLE,
                                          QSS-FIXED-ASset-META-DATA,
                                          QSS-FIXED-ASSET-BUFF

       IF NOT DB-GOOD-STATUS
          IF END-OF-FILE THEN
             GO TO ED-EXIT
          ELSE
             DISPLAY " Return Error     : " DB-RETURN-STATUS
             DISPLAY " qdb_init STATUS  : " DB-OTHER-STATUS
          END-IF
       END-IF
```

Ovr Tab Read Ln 583, Col 7

SQL Results Discussion

- Memory table on client or server (server side cursor) contains results
- Return n-rows of column data into local memory, one column (field) at a time
- Data is ascii readable null terminated (think ‘c’ string) and must be converted to COBOL fields before use in COBOL code

Mapping SQL Data to Buffer

- Characters are simple byte moves
- Date/Time must convert date/time format
- Numeric data must convert numbers like “123.45” to standard COBOL pictures

Making it Possible

- We chose standard data types – Fixed length Char, Date, Time, Numeric
- We decided to avoid binary pictures
- Metadata is programmatically available
- We added our own additional metadata structure for persistence across similar table access (note: we rarely use TurboImage item lists). This structure is filled in by QDBI the first time a table is accessed
- QDBI uses metadata to map from SQL results to COBOL record buffer

Performance Issues

- Converting each field's ascii SQL result to a COBOL buffer is a lot of overhead - 10,000 rows with 70 columns results in 700,000 conversions!
- Typical COBOL report programs for TurboIMAGE will DBGET and then select records by testing selection criteria
- Drop-in replacement of DBFIND with a select that selects all records and still contains logic to select records by testing selection criteria gives you a double performance hit and eliminates the SQL "select" feature of the RDBMS

Mitigating Performance Issues

- Don’t return all columns – return a subset to significantly reduce the number of field conversions
- Move the logic to select records into your SQL select to reduce the number of returned rows
- Run linux on a system with more CPU than your HP e3000

QDBI - Architecting a Migration Solution

- Provide for ‘drop-in’ replacement called routines with extensions to subset the returned columns
- Provide for ‘drop-in’ with the ability to include the selection criteria in the Select instead of in the COBOL logic. Provide for dynamic build of SQL by QDBI by passing appropriate field/criteria/operator values.
- Design ‘drop-in’ to make migration easy
- Use build-in extensions when performance improvements are required (after basic migration has been performed)

Test Results from Pilot

- General feeling is that SQL access takes about 10-12 times more cpu time than similar access using TI
- But... you don't spend all your time in the DB
- Single SQL process accessing 73,000+ rows in multiple selects was slightly faster than same TurboIMAGE process. SQL was P-III with .5gb,IDE; TI was A400-110 (55) with 2gb, SCSI
- Multiple simultaneous tests came out equivalent indicating either MPE or a system configuration (memory, disk) influence on improved performance
- When migrating to linux/ia-32 you will have a substantially faster CPU and this will help mitigate performance issues

PostgreSQL Server Control

- initdb – prepare directory area and template db for new PostgreSQL system
- initlocation – initialize secondary db storage location
- ipclean – clean up orphaned semaphores and shared memory after db server crash
- pg_ctl – control functions (start/stop/etc)
- pg_passwd – manage pwd when using PostgreSQL authentication
- postgres/postmaster – db server engine

PostgreSQL DBA functions

- createdb – create new database
- createlang – register new language to DB
- createuser – add new user to PostgreSQL
- dropdb – delete specified DB
- droplang – removes a language from DB
- dropuser – remove user from PostgreSQL
- pg_dump/pg_dumpall/pg_restore – database export backup and restore
- vacuumdb – reclaims wasted disk and updates profile data for query optimizer

The End…