How To Dump And Load

Sometimes it becomes necessary to reorganize the data in your database (for example, to move data from type i data areas to type ii data areas so you can take advantage of the latest features)or to move parts of it from one database to another. The process for doping this can be quite simple or quite complex, depending on your environment, the size of your database, what features you are using, and how much time you have. Will you remember to recreate the accounts for SQL users? To resotre theie privileges? Will your loaded database be using the proper character set and collations? What about JTA? Replication? etc. We will show you how to do all the other things you need to do in addition to just dumping and loading the data in your tables.

How To Dump and Load

gus bjorklund
head groundskeeper,
parmington foundation

What do we mean by dumping and loading?

- Extract all the data from a database (or storage area)
- Insert the data into a new database (or storage area)
- Could be entire database or part

Why do we dump and load?

Why do we dump & load?

- To migrate between platforms
- To upgrade OpenEdge to new version
- To repair corruption
- To “improve performance”
- To change storage area configuration
- To defragment or improve “scatter”
- To fix a “long rm chain” problem
- Because it is October

Ways to dump and load

- Dictionary
- 4GL BUFFER-COPY
- Binary
- Replication triggers (or CDC)
- Table partitioning / 4GL
- Incremental by storage area

Binary Dump & Load

- binary dump files
  - not "human readable"
  - contain copies of database records, no index information
  - are written in an operating system and cpu architecture independent data format
  - portable amongst all OpenEdge systems
- .bd files are portable *upward* across Progress/OpenEdge versions
  - Not so portable going backward

Dump and load steps

- Collect information
- Practice
- Dump
- Prepare new database
- Load
- Verify new database
- Return to production
- Cleanup

0) Collect information

Collect information

- where is database?
- what are the dba user name and password
- is SQL being used?
- where are the database server startup parameters
- where are server startup scripts or is admin server used?

Collect information

- cd /prodb-tmp/skudb
- mkdir –p gus
- proutil sku –C describe >gus/sku.desc
- proutil demo –C tabanalys >gus/sku.anal
- prostrct list sku; mv sku.st gus
- pro sku –p dumpdf.p; mv sku.df gus
- sqlschema -db sku -S sku –u dba –a 123456 \
  –g %.% -o gus/skuprivs.sql
- save db log file; mv sku.lg gus

dumpdf.p

```
create alias dictdb for database sku.
run prodict/dump_df.p ("ALL",
                       "/prodb-tmp/skudb/gus/sku.df",
                       "").
```

Sequences

- dump sequence definitions via dict (to _seqdefs.df)
- dump sequence values via dict (to _seqvals.df)

Make a detailed checklist

- Every step, every command
- Make custom scripts as needed
- Someone else should be able to do everything by following
- Point is to make sure nothing is forgotten
- Annotate checklist during dump and load to note times and other useful info to make next time easier

[] database to be dumped and loaded is d:\eb2db\ghspeb
there is enough disk space to keep original files
so we do not have to do a "before" backup
[] in D:\eb2db in directory gus are script generator gen.p and related files
dumptables.bat, logs to dump.log
loadtables.bat, logs to load.log
directory named dump
check scripts before using
[] note sufficient disk space in D: and C:
[] check that all scheduled jobs for backups, ai archiving, etc. are off.
turn them off if needed.
[] connect to ghspeb with promon to determine database are not being used
[] note db, bi, ai, block sizes, bi cluster size, db characterr set
[] shut down the database
[] generate a "before" dbanalysis report, save. estimate 2.5 hours. move to gus dir

[] generate new database storage structure file ghspeb.st
[] turn off after-imaging
[] dump all the table definitions, move to gus directory
[] dump all the user accounts, move to gus directory
[] dump sequence values, move to gus directory
[] dump sql privileges, move to gus directory
[] truncate before-image log
[] run dump script, monitor as needed, check for errors. estimate 3.5 hours, maybe 4
[] copy ghspeb.lg to gus
[] move all d:\eb2db\ghspeb* files to directory \eb2db\oldghspeb for safe keeping
[] move all d:\eb2bi\ghspeb* files to directory \eb2db\oldghspeb for safe keeping
[] adjust structure file as needed to make extents uniform and big enough

[] create new structure in d:\eb2db with -blocksize 8192
[] copy %DL%\empty8 to d:\eb2db\ghspeb
[] enable large files
[] load table/index definitions
[] load user accounts
[] load sequence values
[] load sql privileges
[] binary load data files, estimate 1.5 hours, maybe 2
[] rebuild indexes, estimate 3.5 hours
[] generate "after" database analysis report, estimate 1 hour
[] start a full backup, estimate 2 hours
[] compare before and after record counts
[] delete dump files
[] set ai block size, bi cluster size to same values
[] when backup finishes, enable after-imaging

[] start database server
[] turn monitoring back on for ghspeb
[] copy backup to ghmfgpro2 i: drive
[] make sure scheduled tasks are re activated.
[] clean up files and other detritus generated during dump/load
[] save checklist

1) Practice

Practice

- How long will tabanalys take?
- How long will dump take?
- How much disk space will dump files need?
- Do you have room for old and new database?
- Is your new structure file correct?
- How long will load take?
- How long will backups take?
- How will you recover if something goes wrong?
  - How long will that take?
- Make notes of all time estimates

2) Dump

Dump preparation

Dump preparation

- turn off any scheduled batch jobs, etc.
- promon to database; verify no users logged in
  - promon /prodb/skudb/sku
- shut down database
  - proshut /prodb/skudb/dku -by
- make a backup, archive ai files if needed
  - probkup sku /backups/sku/sku_181002.bak

Dump preparation

- get a current structure file
  - prostrct list sku gus/sku.st
- run a fresh table analysis
  - proutil /prodb/skudb/sku –C tabanalys >gus/sku.anal

table analysis

Totals: 721046601 81.2G 6 3200 120 721046891 1.0 1.0

total number of records
(includes schema area)

binary dump

```
proutil demo -C dump customer ./dumpfiles \
          -RO -B 5000 >> dump.log 2>&1
```

binary dump script

```
#! /bin/bash
# binary dump script for Linux.  error checking removed.
#
LOG=dump.log
DB=db/xx
LIST=dumplist.txt
DUMP_DIR=dumpfiles
mkdir -p ${DUMP_DIR}
N=`wc -l ${LIST}`
echo `date +"%H:%M:%S"` ":: starting dump of ${N} tables from ${DB} ..." >>$LOG
N=0
while read tblName
do
    echo `date +"%H:%M:%S"` ":: ${N}: starting ${tblName} ..." >>$LOG
    stime=`date +"%s"`
    proutil ${DB} -C dump ${tblName} ${DUMP_DIR}/ -RO -B 5000 2>&1 >>proutil.log
    N=$((N + 1))
    etime=`date +"%s"`
    echo `date +"%H:%M:%S"` ":: ${tblName} dumped in $((etime - stime)) sec." >>$LOG
done < ${LIST}
echo `date +"%H:%M:%S"` ":: dump of ${N} tables from ${DB} completed." >>$LOG
```

generate table list

```
define stream d.
output stream d to "dumplist.txt".

for each _file where (_file-num > 0) and (_file-num) < 32768.
    put stream d unformatted _file-name skip.
end.
output stream d close.


or:
     let proutil dump do it (see -dumplist option)
```

3) Prepare New Database

prepare new database

- verify updated structure file sku.st, adjust if needed
- prodel /prodb/skudb/sku
  - or rename existing database files (data extents, bi, ai)
  - or leave in place, copy empty over it
- prostrct create sku sku.st –bocksize 8192
- procopy $DLC/empty8 /prodb/skudb/sku (db codepage ???)
- proutil /prodb/skudb/sku -C EnableLargeFiles
- proutil /prodb/skudb/sku -C enablelargekeys
- proutil /prodb/skudb/sku -C enableseq64

prepare new database

- proutil /prodb/skudb/sku –C enableB2
- enable other options if needed (TDE, partitioning, mutitenancy, auditing, etc)
- proutil /prodb/skudb/sku -C truncate bi –biblocksize 16 –bi 256
- pro sku –B 5000 /prodb/skudb/sku –p gus/loaddf.p
- tablemove, indexmove if needed
- load sequence definitions and values via dict
- load users via dict
- load sql privileges
- proutil /prodb/skudb/sku –C describe >gus/newsku.desc
  - verify you set everything that needs to be set

loaddf.p

```
create alias dictdb for database sku.
run prodict/load_df.p ("/prodb-tmp/skudb/gus/sku.df").
```

Load SQL privileges

- cd database sku's home direcotry
- proserve sku–B 5000 –S 12345
- sqlexp sku–S 12345 –user <sqldba> -password ***** \
  < gus/sqlprivs.sql
- proshut sku -by

4) Load

binary load procedure

- load the data
- build indexes
- verify database
- prepare for production use
- cleanup

binary load

```
proutil newdb/demo -C load ./dumpfiles/customer.bd \
          -r -B 5000 >> load.log 2>&1
```

binary load script

```
#!/bin/bash
# binary load specified list of tables to database
#
LOG_DIR=`pwd`/logs
LOG=${LOG_DIR}/load.log
DB=/prodb-tmp/skudb/sku
LIST=loadlist.txt
LOAD_DIR=`pwd`/dumpfiles
ls ${LOAD_DIR} >loadlist.txt
N=`wc -l <${LIST}`
echo `date +"%H:%M:%S"` ":: starting load of ${N} tables to ${DB} ..." >>$LOG
N=0
while read tblName
do
    echo `date +"%H:%M:%S"` ":: ${N}: starting ${tblName} ..." >>$LOG
    proutil ${DB} -C load ${LOAD_DIR}/${tblName}.bd -r -B 5000 2>&1 >>${LOG_DIR}/proutil.log
    N=$((N + 1))
    echo `date +"%H:%M:%S"` ":: finished ${tblName} " >>$LOG
#
done < ${LIST}
echo `date +"%H:%M:%S"` ":: load of ${N} tables from ${DB} completed." >>$LOG
```

After all tables loaded,
build the indexes

proutil sku –C idxbuild -TB 32 –TM 31 –B 5000

5) Verify database

Verification

- proutil newdb –C tabanalys >newdb.anal
  - compare record counts
- proutil newdb –C describe >newdb.desc
  - compare with original

6) Return to production

Return to production

- proutil /prodb/skudb/sku -C truncate bi -biblocksize 8 -bi 8192
- rfutil /prodb/skudb/sku -C aimage truncate -aiblocksize 8
- probkup /prodb/skudb/sku /backups/sku181004_new.bak
- rfutil /prodb/skudb/sku –C aimage begin
- Enable after image archive daemon
- Review startup parameters, adjust if needed
- Start server and related processes (apw, biw, aiw)
- Re-enable any batch jobs that you disabled

7) Cleanup

cleanup

- delete dump files
- delete any other stuff you left lying around (df files, tabanalys, etc)
- delete original database if not done earlier
  - or wait a few days just in case
- save annotated checklist for next time
- get some sleep

ceisteanna