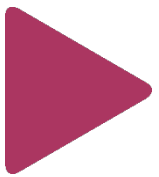
YottaDB™ Foundation

Comsan Channma (Neo)

T.N. Incorporation Ltd.



TN

Agenda

2nd Day.

- Database Extract and Load
- Database Backup and Restore
- Database Integrity Check
- Database Fragmentation
- Database Monitoring

Database Extract and Load

MUPIP EXTRACT

- The EXTRACT command copies specified globals from the current database to a sequential output file
- The format of the EXTRACT command is:

MUPIP> EXTRACT -qualifier file-name

- The optional qualifiers are:
  - -FO[RMAT]=GO | B[INARY] | Z[WR]
  - -FR[EEZE]
  - -LA[BEL]=text
  - -[NO]LO[G]
  - -S[ELECT]=global-name-list
  - -R[EGION]=region-list

MUPIP EXTRACT Example

- Extract the ^ACN global and place it into a binary file named acn.bin:

  MUPIP> EXTRACT -FORMAT=BINARY -SELECT=”^ACN” acn.bin

  ACN Key Cnt: 1659 max rec size: 188

  EXTRACT TOTAL Key Cnt: 1659 max rec size: 188

  The data remains in the database file after the extract has taken place

- Extract the ^ACN global and placed it into a global file acn.go:

  MUPIP> EXTRACT -FORMAT=GO -SELECT="^ACN" acn.go

  ACN Key Cnt: 1659 max subsc len: 18 max data len: 181 max rec len: 188

  EXTRACT TOTAL Key Cnt: 1659 max subsc len: 18 max data len: 181 max rec len: 188

  Here is an example of what the data in the file looks like:

  GT.M MUPIP EXTRACT

  15-APR-1998 23:34:56

  ^ACN(50068,1)

  FIRST LAST|||

  ^ACN(50068,49)

  |2|0||11|1||0||0|1|1|2|7D|||||1|||||1||1|1|15|0|0||0

  ^ACN(50068,50)

  502|L|LN||1|FIRST LAST,FIRST |0|0||||RM|||||||500A

MUPIP LOAD

- The LOAD command enters global variable names and their corresponding data values into a YDB database from a sequential file in one of three formats:
  - GO (global output)
  - BINARY (binary format)
  - ZWR (ZWrite)

  By default, LOAD uses FORMAT=ZWR
- This command uses the Global Directory to determine which database file to use. The LOAD command may operate concurrently with normal YDB database access

MUPIP LOAD (Cont.)

- The format of the LOAD command is:

  MUPIP> LOAD -qualifier file-name

- The optional qualifiers are:
  - -FO[RMAT]=GO | B[INARY] | ZWR
  - -BE[GIN]=integer
  - -E[ND]=integer
  - -FI[LLFACTOR]=integer

MUPIP LOAD Example

- Load in a binary file named acn.bin. (Only fill 85% of each data block within the database file.)

```
MUPIP> LOAD -FORMAT=BINARY -FILL=90 acn.bin

        Label = GDS BINARY EXTRACT LEVEL

        21998041523413002048005100051OGT.M MUPIP EXTRACT

        LOAD TOTAL          Key Cnt: 1659  Max Subsc Len: 18  Max

        Data Len: 181

        Last LOAD record number: 64
```

Database Backup and Restore

Backup Scenario

- What?
  - DB files. (with GLD file and other related files)
  - Working path.
  - Source code.
  - Binary.
- When?
  - Daily, Monthly
- How?
  - Copy files
  - Snapshot

MUPIP BACKUP

- The BACKUP command copies from one or more Greystone Technology Database Structure (GDS) files to a new file or files. This command suspends updates to all regions specified by the BACKUP command from the time it starts the first region until it finishes the last region
- This ensures that BACKUP captures a consistent application state. The command does not suspend processes that only perform retrievals

MUPIP BACKUP (Cont.)

- The format of the backup command is:
  - MUPIP> BACKUP [-qualifier] region-list file-name
  - Qualifiers to the backup command include:
    - -DATABASE -- Creates a disk-to-disk backup copy of the files of all selected regions.
    - -Bytestream -- Transfers MUPIP BACKUP output to a TCP connection, file, or a pipe.
    - -INCREMENTAL – specifies that backup copy only changed blocks
    - -[NO]ONLINE -- allows updates to the database while backing it up
    - -[NO]NEW – creates a new set of journal files for the backup
- Important notes on backing up database files includes:
  - To BACKUP only one region, the file name must resolve to a Unix file or directory name
  - To backup several regions, the file-name must be a directory
    - If the file-name is a directory, MUPIP assigns the backup files the same name as the file associated with the dynamic segment of each region. The target directory, therefore, must not contain any of the regions included in the BACKUP
  - Incremental backups have to be restored to the database with the mupip restore command. Comprehensive backups can be done using the Unix cp command
  - Use of "*" to reference all db files

MUPIP BACKUP Example

- To backup the file associated with the region UBG to the file mumps.bak:

```
MUPIP> BACKUP UBG mumps.bak

DB file /ydbinst/mumps.ubg backed up in file mumps.bak
```

  - DB file /ydbinst/mumps.ubg backed up in file mumps.bak. By default, the backup was comprehensive and no new journal files were created. This command requires stand-alone access and restricted updates to the db file.
  - To restore the backup file, use the Unix copy command to copy the file back to it's original destination.

```
$ cp mumps.bak mumps.ubg
```

MUPIP BACKUP Example (Cont.)

- To backup all files with new journal files in a directory named /BACKUPS_DIR without interrupting users access

```
MUPIP> BACKUP "*" -online -new /ydbinst/BACKUPS_DIR
```

  - A comprehensive backup of all db files was created in the /ydbinst/BACKUPS_DIR with a corresponding set of journal files while allowing updates to the db files

- To backup all updates to all files in a directory named /BACKUPS_DIR.

```
MUPIP> BACKUP  -incremental "*" /ydbinst/BACKUPS_DIR
```

  - A backup of the updates to all the database files since the last backup was created in /ydbinst/BACKUPS_DIR
  - To restore the backup file, the mupip restore command must be used

MUPIP BACKUP Example (Cont.)

- The MUPIP BACKUP command can also use system utilities:
- MUPIP BACKUP can be of the form "| <string> where <string> is executed and the backup output is passed in as standard input. The backup qualifiers can be freely mixed with any/all of the following commands.
  - mupip backup "*" "|gzip -c"
    - creates a compressed backup of all databases
  - mupip backup "*" tcp://pharaoh:5000
    - sends the backup to the machine pharaoh at port 5000, assuming a listener is awaiting the output

MUPIP RESTORE

- Integrates one or more BACKUP -INCREMENTAL files into a corresponding database.
- The transaction number in the first incremental backup must be one more than the current transaction number of the database.
- The format of the RESTORE command is:

  MUPIP> RE[STORE] file-name bytestrm-bkup-list

- Example

  $ mupip restore backup.dat backup.bk1,backup.bk2

Database Integrity Check

MUPIP INTEG

- The INTEG command performs an integrity check on a GDS database file. The command operates on one or more regions in the current global directory by suspending current updates to those regions
- The INTEG command should be done at the following times:
  - Periodically -- to ensure ongoing integrity of the database(s)
    - Frequent INTEGs help catch integrity problems before they spread throughout the database file.
  - After a Crash -- to ensure that the database was not corrupted
  - When Database Errors are Reported -- to troubleshoot the problem

MUPIP INTEG (Cont.)

- The format of the INTEG command is:

  MUPIP> INTEG [-qualifier] -FILE file-name

  MUPIP> INTEG [-qualifier] -REGION region-list

  - The file-name or region-list identifies the target of the INTEG.
  - The INTEG command must include -FILE or -REGION qualifiers that determine whether the argument of the INTEG is a file-name or region-list.
  - The optional qualifiers which determine the action(s) for the SET are:
    - -FAST only looks at index blocks (dramatically faster than full)
    - -FULL looks at index and data blocks (default qualifier)
    - -SUBSCRIPT specifies which global to verify

MUPIP INTEG Example 1

A full integrity check for region UBG:

$ mupip integ –region UBG

Integ of region UBG

No errors detected by integ.

Type	Blocks	Records	% Used	Adjacent
Directory	4	182	29.711	NA
Index	345	13079	32.254	7
Data	12913	569725	72.283	8870
Free	16738	NA	NA	NA
Total	30000	582986	NA	8877

MUPIP INTEG Example 2

A full integrity check on the global ^DBTBL with the -FULL qualifier to show how much space this particular global is taking up:

$ mupip integ -full -subscript="^DBTBL" -region TBLS

Integ of region TBLS
Directory tree

Level	Blocks	Records	% Used	Adjacent
1	1	1	0.781	NA
0	1	7	5.175	NA

Global variable ^DBTBL

Level	Blocks	Records	% Used	Adjacent
3	1	2	3.759	0
2	2	146	85.571	1
1	146	9212	52.288	2
0	9212	274689	66.460	6751

No errors detected by integ.

Level	Blocks	Records	% Used	Adjacent
Directory	2	8	2.978	NA
Index	149	9360	52.409	3
Data	9212	274689	66.460	6751
Total	15000	284057	NA	6754

Database Fragmentation

MUPIP REORG

- The REORG command is used to defragment and compact database files. The REORG runs concurrently with other database activity, including updates. REORG optimizes the structure of database files but does not handle native file system fragmentation.
- The format of the REORG command is:

```
MUPIP>REORG [–qualifier]
```

REORG Command Qualifiers

- **EXCLUDE-** Restricts swapping blocks used by the specified globals. EXCLUDE will bypass the blocks containing the global/globals specified. Arguments for the EXCLUDE qualifier are:
  - A Global name, such as ACN
  - A range of global names, such as A7:B7
  - A list, such as A,B,C
  - Global names with the same prefix, such as TMP*
- **FILL_FACTOR-** Specifies the percent to fill each database block. Updates to the block fill the remaining available space.
- **RESUME-**If the REORG is stopped, the resume qualifier allows you to restart the REORG operation from the point where the operation stopped
- **SELECT-** By default, REORG operates on all globals in all database files identified by the current global directory. SELECT specifies specific globals to run the REORG operation on. Arguments for the SELECT qualifier are:
  - A Global name, such as ACN
  - A range of global names, such as A7:B7
  - A list, such as A,B,C
  - Global names with the same prefix, such as TMP*
- **REGION -** Specifies that REORG operate in the regions in the associated list and restricts REORG to the globals in those regions that are mapped by the current global directory;

MUPIP REORG Example

- To reorg globals ACN and CIF with a fill factor of 85%:

```
$ mupip reorg -fill_factor=85 -select=ACN,CIF
```

```
Fill Factor: Index blocks 85%: Data blocks 85%
Global: ACN
Blocks processed    : 1473
Blocks coalsced     : 189
Blocks split        : 640
Blocks swapped      : 1465
Blocks freed        : 8
Blocks reused       : 659
Blocks extended     : 0
Global: CIF
Blocks processed    : 86
Blocks coalsced     : 52
Blocks split        : 17
Blocks swapped      : 86
Blocks freed        : 0
Blocks reused       : 17
Blocks extended     : 0
```

Database Monitoring

YottaDB Log

- YottaDB runtime message
- YottaDB process's log

YottaDB Messages

- The YottaDB run-time system sends messages to the system log. These are not trapped by the application error trap.
- Compilation errors generated by YottaDB are directed to STDERR. These are not trapped by the application error trap. You can avoid them by compiling application code before deploying it in production or log them by running mumps processes with STDERR directed to a file.
- Application error are handled by application.
- YottaDB sends messages to the system log at the LOG_INFO level of the LOG_USER facility
- rsyslogd configuration

```
user.info   /var/log/user.log
```

YottaDB Message Severity

- **-I-** for informational messages
- **-W-** for warnings
- **-E-** for errors
- **-F-** for events that cause a YottaDB process to terminate abnormally.
- Messages and Recovery Procedures Reference :
  https://docs.yottadb.com/MessageRecovery/index.html

YottaDB Process Log File

- .mje and .mjo files : STDERR and STDOUT of YottaDB JOB command
- YDB_FATAL_ERROR.ZSHOW_DMP_*.txt : YottaDB process terminates abnormally

YottaDB Database Size

- Database file size
- Data size
- Global size

Database File Size

- OS file size
- ls -l [database file]
- How different of “ls” VS “du” command?

Data Size

- YDB>D ^%FREECNT

```
YDB> d ^%FREECNT
Region          Free     Total          Database file
------          ----     -----          -------------
DATA              94       100 ( 94.0%) /ydbdir/gbls/mumps.data
OCTO              56       100 ( 56.0%) /ydbdir/gbls/mumps.octo

YDB>
```

Global Size

- “Mupip size” command
- Mupip size : Estimates and reports the size of global variables using a format that is similar to the one that appears at the end of the MUPIP INTEG -FULL report.
- The format of the MUPIP SIZE command is:

MUPIP> SI[ZE] [-h[euristic]=estimation_technique] [-s[elect]=global-name-list] [-r[egion]=region-list] [-a[djacency]=integer]

- Example

$ mupip size -heuristic="impsample,samples=2000" -select="y*" -region=”DATA"

Database File Header

DSE> dump -fileheader

DSE> dump -fileheader -all

```
File    /home/jdoe/.yottadb/r1.20_x86_64/g/yottadb.dat
Region  DEFAULT
File            /home/jdoe/.yottadb/r1.20_x86_64/g/yottadb.dat
Region          DEFAULT
Date/Time       27-JAN-2014 03:13:40 [$H = 63214,11620]
  Access method                          MM  Global Buffers                1024
  Reserved Bytes                          0  Block size (in bytes)         1024
  Maximum record size                   256  Starting VBN                   513
  Maximum key size                       64  Total blocks             0x00000065
  Null subscripts                     NEVER  Free blocks              0x0000005E
  Standard Null Collation             FALSE  Free space               0x00000000
  Last Record Backup     0x0000000000000001  Extension Count                100
  Last Database Backup   0x0000000000000001  Number of local maps             1
  Last Bytestream Backup 0x0000000000000001  Lock space               0x00000028
  In critical section            0x00000000  Timers pending                    0
  Cache freeze id                0x00000000  Flush timer             00:00:01:00
  Freeze match                   0x00000000  Flush trigger                   960
  Freeze online                       FALSE  Freeze online autorelease    FALSE
  Current transaction    0x0000000000000006  No. of writes/flush              7
  Maximum TN             0xFFFFFFFF83FFFFFF  Certified for Upgrade to         V6
  Maximum TN Warn        0xFFFFFFFD93FFFFFF  Desired DB Format                V6
  Master Bitmap Size                    496  Blocks to Upgrade       0x00000000
  Create in progress                  FALSE  Modified cache blocks            0
  Reference count                         1  Wait Disk                        0
  Journal State                    DISABLED
  Mutex Hard Spin Count                 128  Mutex Sleep Spin Count          128
  Mutex Queue Slots                    1024  KILLs in progress                0
  Replication State                     OFF  Region Seqno    0x0000000000000001
  Zqgblmod Seqno         0x0000000000000000  Zqgblmod Trans  0x0000000000000000
  Endian Format                      LITTLE  Commit Wait Spin Count           16
  Database file encrypted             FALSE  Inst Freeze on Error          FALSE
  Spanning Node Absent                 TRUE  Maximum Key Size Assured       TRUE
  Defer allocation                     TRUE  Spin sleep time mask    0x00000000
  Async IO                              OFF  WIP queue cache blocks            0
  DB is auto-created                  FALSE  DB shares gvstats                TRUE
  LOCK shares DB critical section     FALSE
```

Database File Header Element

- Block size (in bytes) : The size (in bytes) of a GDS block.
- Extension Count: The number of GDS blocks by which the database file extends when it becomes full.
- Global Buffers: The number of BG buffers for the region.
- Maximum key size: The minimum key size is 3 bytes and the maximum key size is 1019 bytes.
- Maximum record size: The minimum record size is zero. The maximum is 1,048,576 bytes (1MiB).
- Freeze match: 0x00000000 mean DB is not suspend, others mean DB is suspended

Question and Answer