OpenTURBO A.01.04 Known Issues:

2003-02-04:

QUERY.PUB.SYS, you must specify a valid local TurboIMAGE database, which is also needed to be named the same in the CONFIG file for a OT_DUALMODE access.

OpenTURBO A.02.00 Known Issues:

2003-02-04:

OT_DUALMODE = ON, if the database password ‘;’ is used, then DMDRV.BIN.IMAXSOFT must be launched from TurboIMAGE database’s creator.

2003-02-04:

DBACCESS and DBLOCK will add DBASE to it, for multiple-context support. Currently, one DBCLOSE will close all concurrent DBOPENs to the same logical TurboIMAGE database in both libsqlD and libsqlD02 SQL Direct Access Mode libraries.

2003-06-13: A.02.00 and A.02.01 will take care of this issue.

2003-02-10:

TurboIMAGE K is mapped to ORACLE RAW, which is not yet supported. As of 2003-02-11, K to RAW is supported.

2003-02-11:

Challenging points:

- How do you handle ORACLE context and XA?
- How do you handle ORACLE transaction?
- How do you handle ORACLE locking?
- How do you handle ORACLE cursors?
- Static or dynamic query, block or single fetch, pre-fetching, caching, . . . etc.?
- K to RAW, how do you handle RAW?
- Data mapping? Packed decimal? Binary data in X?

2003-02-12:

Automatic Dataset:

- No need to exist or to migrate at all - significantly performance saving, it is ORACLE index.
- The reasons to migrate A dataset is that your apps access A Dataset directly via DBGET (mode 2, serially) or DBINFO.
- With A dataset, D dataset INSERT trigger will insert A dataset automatically, but not for DELETE. So, you need to run otCLEAN to clean-up dangling keys from A datasets.
- DOOR - no need to replicate A datasets, D dataset INSERT trigger will insert A dataset automatically, otCLEAN needs to run periodically also.

2003-06-08:

Southwest OPEN ISSUES (06-04-2003 thru 06-06-2003):

Date: June 8, 2003

From: iMaxsoft Corp.
For: Southwest Airlines On-site Training

Target Accomplishments:

1) HP3000 PETI and PETB environment unload scripts, which include PNRDB.BASEQ802, REFDB.BASEREF, and BKODB.BASEBKO 3 databases in PETI account for UPDATE testing, and PNRDB.BASEQ203, PNRDB.BASEQ303, and REFDB.BASEREF 3 databases in PETB account for READ testing.

   Status: done.

2) HP3000 PETI and PETB 6 databases unload. The total size of those 3 major PETB databases is 55.57G. The database unload summary, for all but Automatic sets, all paths for all detail sets (preserve all chronological path orders), organize all sets via their TurboIMAGE internal record-num, is

```
PNRDB.BASEQ203.PETB (13.30 hours),
PNRDB.BASEQ303.PETB ( 8.15 hours),
REFDB.BASEREF.PETB  (12.05 hours).
```

   Status: done for all manual and detail datasets but one HP-LOG from REFDB.BASEREF.PETB. At the last step of HP-LOG unload process, OpenTURBO couldn't create new data file; this is caused by exceeding libc's byte-file size limit, HP-LOG file record count is 27,918,856 and record size if 94 bytes, so the equivalent byte-file size is 94*21,918,856 = 2,624,372,464 > 2^31=2,147,483,648. We can either ftp our staging file to HP-UX and complete the final step on HP-UX directly, or use HP3000 HPFOPEN(). It will be fixed in a week.

   Revised Status: Done (2003-06-18)

   Bug fixed 2003-06-18 by using HPFOPEN, FWRITE, and FCLOSE.

```
HP-LOG with –c      => 4.5 hours
HP-LOG without –c   => 19 minutes
```

3) ORACLE PETIIMAX and PETBIMAX environment load scripts, which include accounts OT, PNRDB_BASEQ802_PETI, REFDB_BASEREF_PETI, BKODB_BASEBKO_PETI, PNRDB_BASEQ203_PETB, PNRDB_BASEQ303_PETB, and REFDB_BASEREF_PETB, and SQL/DDL for setup, access privileges, triggers, sequences, tables, indexes, etc.

   Status: done.

4) ORACLE PETIIMAX environment creation and data load, which include accounts OT, PNRDB_BASEQ802_PETI, REFDB_BASEREF_PETI, and BKODB_BASEBKO_PETI.

   Status: done.

5) Test cases:

   1. DOOR Testing:

      a. scripts setup: done
      b. COBOL2 and TurboIMAGE (XL.PUB.SYS): done (06-19-2003)

      Issues:
      __________ Multi-TurboIMAGE or single-TurboIMAGE logfile

Current DOOR supports single-TurboIMAGE logfile only.

07-04-2003 Offline Data Replication without TurboIMAGE DB. Use intercam and shootam programs.

2. COBOL2 Testing:

a. COBOL2 and TurboIMAGE (XL.PUB.SYS) : done
b. COBOL2 and TurboIMAGE (OpenTURBO) : done
c. COBOL2 and ORACLE (OpenTURBO) : 06-12-2003
d. COBOL2 and ORACLE in DUAL-MODE : 06-24-2003

Issues:

YES COBOL2 in ANSI mode; cob85xl is ANSI85 compliant.

NO OpenTURBO special version for TurboIMAGE call parameter boundary checking and re-alignment (will have some performance penalty and should be used for development only) – if this can speed up your POC process, I can raise its priority. (use COBOL's $CONTROL BOUNDS or VALIDATE)

2003-06-09:

HP-UX listener core dump:

If the CONFIG file is not specify during launching 'listner', our listner cd its home group to /tmp, since we didn't start the listner via k, so it doesn't have the proper privilege, hence there is no core file created.

We should always use CONFIG and specify the HOME, then any core goes to the HOME, but if you have multiple aborts, one will override the other, our next release will address this rundir issue.

2003-06-12:

Summary of Southwest POC:

1) DOORMAP and DRMAPAM (AMISYS and OFFLINE)
   -i TIFile
   -m Map Report
   -p Port Name or Number
   -c Y/N to –c
2) DOOR: Multi-database Logfile
3) DOOR: Offline Logfile Replication
4) DOOR: Offline Logfile Replication without TurboIMAGE
5) OTXL: OT_DUALMODE = WRITE (ignores READ comparison, for OLTP apps)
6) OTXL: OT_DUALMODE = ON (READ/WRITE comparision, for BATCH apps and reports)
7) OTXL: OT_DUALMODE TurboIMAGE Primary (OTB_BE_MPE)
8) OTXL: OT_DUALMODE ORALCE Primary (NOT OTB_BE_MPE)
9) OTXL: Byte, 2-Byte, 4-Byte and 8-Byte alignment during data movement (move data into TurboIMAGE data buffer)
10) OTDRV: rename OTDAT files with extension .DAT
11) OTDRV: rename OTCLT files with extension .CLT
12) OTDRV: rename OTSCHEMA files with extension .sql
13) CONFIG file:

```
OT_RDB_OWNER        = AMISYS   (-eCROSSREF)
OT_RDB_SID          = TRIPPOC  (reserved for future)
```

```
OT_RDB_LOGON        = db_grp_acct/xx@SQLNet
OT_IMAGEMODE        = ON        (supersedes OT_DUALMODE)
                                FALSE – 2003-06-21
OT_AUTO_COMMIT      = ON        (used for DBOPEN mode 3&4)
OT_BTREE            = ON        (emulate DBUTIL/BTREE enable)
```

2003-06-21:

HP3000 MPE/XL:

```
OT_IMAGEMODE = ON          TurboIMAGE is the prime
OT_DUALMODE  = OFF         Access local TurboIMAGE only

OT_IMAGEMODE = ON          TurboIMAGE is the prime
OT_DUALMODE  = ON          Dual mode is using TurboIMAGE
                           as the primary database and
                           Oracle as the secondary

OT_IMAGEMODE = OFF         ORACLE is the prime
OT_DUALMODE  = ON          Dual mode is using ORALCE as
                           the primary database and
                           TurboIMAGE as the secondary
```

HP9000 HP-UX:

```
OT_IMAGEMODE = ON          TurboIMAGE is the prime
OT_DUALMODE  = OFF         Access remote TurboIMAGE only

OT_IMAGEMODE = ON          TurboIMAGE is the prime
OT_DUALMODE  = ON          Dual mode is using TurboIMAGE
                           as the primary database and
                           Oracle as the secondary

OT_IMAGEMODE = OFF         ORACLE is the prime
OT_DUALMODE  = ON          Dual mode is using ORALCE as
                           the primary database and
                           TurboIMAGE as the secondary

OT_IMAGEMODE = OFF         ORACLE is the prime
OT_DUALMODE  = LOG         Dual mode is using ORALCE as
                           the primary database and
                           TurboIMAGE as the secondary;
                           TurboIMAGE calls are not
                           performed in real-time mode,
                           all WRITE info are logged to a
                           logfile for offline TurboIMAGE
                           recovery.
```

Special for DBPUT():

```
OT_DUALMODE        = ON
OT_USE_IMAGERECNUM = ON    OpenTURBO does TurboIMAGE
                           DBPUT() first and uses the
                           returned RecNum for ORALCE
                           INSERT statement. This is to
                           ensure the exact mirroring of
                           TurboIMAGE internal to ORACLE.
```

Special for OT_DUALMODE:

```
OT_DUALMODE = OFF/ON/WRITE/LOG

   OFF     Disenable DUAL-MODE
   ON      DUAL-MODE for Read and Write comparison
   WRITE   DUAL-MODE for Write comparison only
```

LOG DUAL-MODE for MPE-IMAGE Fail-over only

14) OTXL: Logging and Recovery to TurboIMAGE
15) Installation: HP-UX relative path and share library path
16) otgenDDL: same as ALLBASE/SQL sqlgen

```
otgenDDL –u –x –o –t –s

  usage: otgenDDL -u -x -o -t -s

    -u ORACLE Login User/Encrypted Password
    -x genCreateTABLE, genCreateINDEX, genCreateTRIGGER,
       genCreateSEQ,   genCreatePRIV,  genCreateCONSTRAINT,
       genDropTABLE,   genDropINDEX,   genDropTRIGGER,
       genDropSEQ,     genDropPRIV,    genDropCONSTRAINT,
       genALL,
       genIMAGESchema, chkIMAGESchema,
       validateOT,     syncOT,
       genTIFILE,      genCHANGE
    -o Owner Name (wildcard % is allowed)
    -t Table Name (wildcard % is allowed)
    -s Output Script Filename

GENCREATETABLE       = by owner, table and wildcard
GENCREATEINDEX       = by owner, table and wildcard
GENCREATETRIGGER     = by owner, table and wildcard
GENCREATESEQUENCE    = by owner, table and wildcard
GENCREATECONSTRAINT  = by owner, table and wildcard
GENCREATEPRIVILEGE   = by owner, table and wildcard
GENDROPTABLE         = by owner, table and wildcard
GENDROPINDEX         = by owner, table and wildcard
GENDROPTRIGGER       = by owner, table and wildcard
GENDROPSEQUENCE      = by owner, table and wildcard
GENDROPCONSTRAINT    = by owner, table and wildcard
GENDROPPRIVILEGE     = by owner, table and wildcard
GENALL               = Generate all create and drop statements

GENIMAGESCHEMA       = by TurboIMAGE DBName and wildcard
CHKIMAGESCHEMA       = TurboIMAGE schema syntax checker

VALIDATEOT           = Validate OT Tables, TIFile and Checksum
SYNCOT               = Reset OT tables contents via new TIFile
                       and reset Checksum

GENTIFILE            = Create TIFile via old TIFile and new
                       TurboIMAGE schema (TILOAD accesses
                       TurboIMAGE root-file directly)
GENCHANGE            = Generate discrepancy SQL commands for
                       ORACLE structure change
```

2003-06-13:

Need from Beta:

1) DBIOMAIN CALL statement and its LINKAGE section definition
2) Migrated ORACLE database schema, including all ORACLE objects, such as indexes, triggers, sequences, constraints, . . .

How to plug-in OpenTURBO:

1) Use the AMISYS migrated ORACLE database as it (NO RE-BUILD or NO RE-LOAD)

2) Use OpenTURBO script to add additional triggers, indexes, and other necessary ORACLE objects; a clean-up script is also provided for dropping all OpenTURBO created objects.
3) Replace AMISYS share library with OpenTURBO libot.sh; the standard practice is to use SHLIB_PATH= and symbolic link to point to it.
4) READY for testing . . .

Summary of OpenTURBO pros and cons:

Pros:

1) Lean core technology.
2) Off-the-shelf supported product.
3) Migration roadmap - emulator, translator and native apps.
4) Easy to learn, deploy and distribute.
5) Vendor Domain knowledge – HP-UX, ORACLE, TurboIMAGE, MPE/XL, and other open platforms.
6) Dynamic SQL Query solution, it is easy for performance enhancements and functionality enhancements.
7) Phases or staged migrations are supported.
8) Rollback/Recovery to HP/3000 is guaranteed for both apps and databases.
9) Open source for MPE/XL system intrinsic routines.
10) Open source for MPE/XL CI command emulator.
11) Open source for COBOLII to ANSI profiler.
12) Bi-directional and DUAL-MODE data and apps mirroring between TurboIMAGE and ORACLE.

Cons:

1) Proprietary technology.

Summary of TRANSOFT's solution:

Solution provides:

1) TurboIMAGE Intrinsic to ORALCE Pre-processor SQL source code generation.
2) COBOLII to MFCOBOL migration; alignment for internal and external (foreign language) calls, COBOLII specific pseudo-intrinsic (such as .LEN.), and etc.
3) MPE/XL Intrinsic migration; MPE system intrinsic to HP-UX system intrinsic conversion, such as HPFOPEN, GETINFO, etc.
4) MPE/XL CI and JCL migration; MPE CI command to HP-UX shell command conversion, SUPRTOOL and COGNOS job script conversion (data-item, syntax, and flow-control), and JOB flow-control conversion (STREAM command and SPOOLER manager).

TurboIMAGE Intrinsic to ORACLE Pre-processor SQL source code generation is the core technology that drives AMISYS's entire apps, below is a summary of Transoft's core technology concerns:

1) In-complete item types support: only supports X, I1, I2, J1, and J2, which is a small subset of TurboIMAGE item types, E, R, J, U, Z, P, K, I4, J4, etc.
2) DBLOCK on unknown value for mode 5 and 6: use the previous DBFIND key value? This random key-value may never be set (for example, your program never calls DBFIND, it calls DBGET mode 2/3 instead; even your program does call DBFIND, but the key-value is not necessary the one for DBLOCK), so your subsequent DBUPDATE or DBDELETE will update rows that are un-related to the locked rows; this scenario causes 2

problems, first you lock something that are not suppose to which causes unnecessary lock contention, second, the default ORACLE lock mechanism is triggered for your updates and your process may cause lock conflict to itself.

3) NO PARSE OF LOCK DESCRIPTOR: DBPUT will always be ‘no lock in affect’ for mode 5 and 6, because your DBLOCK will always lock on incorrect key-value; by design, DBLOCK locks non-exist key-value for DBPUT, whereas ORACLE only locks on existing rows, lock on non-exist key-value is not allowed in ORALCE due to its physical lock implementation; hence you have to use set or base level lock (mode 1,2,3 or 4) – strong contention.

4) ENTITY LEVEL LOCK doesn’t work: even you use ENTITY lock in your apps, you still have serious data integrity problem; for example, while you lock one CLAIM and then process its all related updates; unless all sub-entities are 100% dependents to the CLAIM, otherwise you will encounter serious integrity issue no matter what; for example, PROVIDER is not a 100% dependent to the locked CLAIM (ENTITY level), actually PROVIDER is an independent ENTITY and can be linked to many CLAIMs, while you update PROVIDER’s info, other CLAIMs may also update it concurrently, which is a conflict to your CLAIM ENTITY lock tree.

5) Use SET level lock will ease item 4) issue, but it is a very strong and expensive lock in ORACLE environment, and it is not recommended by ORACLE; it significantly reduces your OLTP throughputs, halts you apps easily and is very difficult to locate the problem in a multi-users on-line environment, because your apps are not aborted, they simply sit on a LONG-WAIT state, and you have to find where in you apps ‘missing a commit’ or ‘wait for hit ENTER to continue’.

6) No path chronological orders are preserved, and no original TurboIMAGE record number is migrated. The original TurboIMAGE record number is very useful for debugging initial data migration and data clean-up processes.

7) Commit/Rollback at statement level, Commit/Rollback during DBUNLOCK, Commit/Rollback during DBEND, DBXEND, DBXUNDO? Where and when do you commit? Or do you commit at all? Remember, you don’t commit, no LOCK will be released, and no other processes can access your updated rows.
   No strategy for ‘TRANSACTION’; basically when you issue an update, you have no idea whether you are updating the one you just got or over writing somebody else updates. Transaction is the key for data integrity

8) TurboIMAGE reads un-committed data (DIRTY READ), while ORALCE reads committed data only (RR or RC); in ORACLE, you can access un-committed data in the same process, but not cross processes, which may cause major problem to your apps, it you reply on other process’s updates as the input of your process.

9) NO PARSE OF ITEM-LIST and NO CACHE OF PARSED ITEM-LIST: supports ‘@;’ only item-list; serious performance degradation, it causes unnecessary large volume data movement and conversions between ORACLE server and client processes, and between ORACLE client and your apps (DBIOMAIN and SET-Module).

10) No built-in trace and debug mechanism for problem solving; when problem occurs, there is no mechanism to trace apps and data flows, and to precisely point out where the problem is.

11) Huge share-library and large volume of SQL Host Variables in SET-Modules consume extensive main memory, swap space and system resources; regardless of what you need in your program, you have to pay the full-price period. It is a

serious resources drain and you need MORE CPU, MORE MEMORY, MORE DISC, MORE ORACLE, MORE HP-UX PROCSSES, and etc.

12) Context control (Channel), whether you use multiple connects or not, the penalty is that all HOST VARIABLE ARRAY will be allocated both in you apps and its corresponding ORACLE server process. You may need a 7410/8-ways with 48GB MM and 500GB Disc (huge swap space for on-line user’s heap, 128MM per users * 500 = 64GB) for a 250,000 members plan.

13) Missing 24X7 Migration Methodology (required procedure for a 24X7 operation):

   A. Stop HP/3000 apps and perform a Full Backup.
   B. Unload TurboIMAGE data to files (OpenTURBO/UNLOAD).
   C. Resume HP/3000 apps with TurboIMAGE logging enabled.
   D. Build ORACLE and load data from B) data files.
   E. Replicate data from TurboIMAGE log files to ORACLE (OpenTURBO/DOOR) – this step you can do it whether your apps are one-line or off-line, or even on a separate HP/3000, it accesses the TurboIMAGE log files only.
   F. Stop HP/3000 apps.
   G. Replicate residual data from TurboIMAGE log files to ORACLE (OpenTURBO/DOOR) – this is the last step before you switch to HP-UX, during this period, you must have your apps completely shut down.
   H. Start HP-UX apps.

   The window A through B is how long you need to shut down your production for data unloads. This process needs to be done once on production. Our benchmark is 13.5 hours for 1TB fully-loaded (all paths, all sets including automatic, and use internal record-number) TurboIMAGE unload.

   Process D and E are keeping your ORACLE up-to-date. You can keep this process running for a while, so that you have enough lead-time for verification.

   To switch to HP-UX and ORACLE, perform F through H, it may take from a few minutes to hours, depends upon the last un-flushed log-file size.

14) Maintenance:

   A. Too much source code to maintain (DBIOMAIN and SET-Modules).
   B. One SET-Module change requires re-build the entire shared library which requires re-test the entire apps.
   C. One SET-Module change requires re-build and re-test customer’s entire surrounding apps and their own-shared library.
   D. If the generator changes its rules or adds new features, you need to re-generate all SET-Modules, re-build apps and re-test for base product as well as surrounding apps.
   E. If you decides to use the generated code as the base for future modifications, then the maintenance efforts would be . . . ?, depends on your in-house expertise.
   F. If you own the generator and its source code, you are in better position, but support from AMISYS and TANSOFT are questionable . . .; you touché it, you are on your own?

Conclusion,

It is always true to isolate the database access routines; they need to be lean, robust and sharable. That’s share-library and COBOL copy-lib are for. That’s what we have in the AMISYS 3000 today. Transoft’s solution is violating the fundamental application design rule. It creates 3 to 5 times more code than original, and lot more if they fix above issues.

This is a very expensive intern-solution, it is so expensive that you risk your entire business on such “FRAGILE” technology, and you know it will break, but you don’t know when and where. “IF YOU (TRANSOFT) KNEW THE POTENTIAL ISSUES, WHY DIDN’T YOU SOLVE THEM IN YOUR INITIAL DESIGN? . . .”, quoted from my original source.

This is a very very specific solution for a very very specific environment, most of the concerns raised above are common and must-solve for all software. “IT IS HARD TO BELIEVE . . . ANY CUSTOMER IS ADOPTING THIS TECHNOLOGY, . . . “, quoted from my original source.

2003-06-17:

-888 for IMAXSOFT13_PATH_01, . . ., etc.

1) for DBUPDATE only.
2) if OT_CIUPDATE = ON, -888 is passed for IMAXSOFT13_PATH_01, which forces update trigger to examine whether updated key exists in the proper Automatic dataset(s) or not, INSERT one if doesn’t.
3) if OT_CIUPDATE = OFF, since update key values are prohibited, so there is no need to pay overhead for examining Automatic dataset(s), so a 0 value is passed for IMAXSFOT13_PATH_01, . . ., etc.

2003-06-18:

ORACLE White Spaces issue:

VARCHAR2,	the white spaces are countable
CHAR,	the white spaces are not countable

2003-07-04:

HP-UX:

MSG:	Message PIPE file support
CIO:	Circular file support
MAP:	Short and long map file support

2003-08-13:

Southwest Airlines trip:

1) DBLOCK() mode 5 and 6 is not implemented in A.02.00, it serializes all DBLOCK() requests, has been fixed in A.02.00 and roll-in A.02.01.
2) DBLOCK() produces too many latching due to table level exclusive lock and commit work. DBLOCK2 only for SWA and use HP e3000 local ALLBASE/SQL.
3) INSERT should be cached.
4) WHITE SPACE, first, uses VARCHAR instead of CHAR, second, data needs to be loaded without white spaces for X, U, and Z; OpenTURBO needs to trim spaces for INSERT and

UPDATE and KEY-IEM values, to fill spaces for X, U, and X for DBGET buffer.

2003-08-23:

TIDRV Overhead:

1) If you use TIDRV for performance benchmarking, the standard overhead is about 40%, it is a little bit less of your use PRINT OFF command to turn off output logging.

DBGET Mode 2 or 3 Serially:

1) In order to produce consistent output, ORDER BY IMAXSFOT13_SEQ_NO is used in the query, it has some performance impact at the beginning of sequential data access.
2) Use dataset level OT_SERIAL_ORDERBY = OFF to turn it off. Default is ON and is supported in A.02.01.

2003-09-03:

New performance options in CONFIG:

1) Database Level:

```
OT_NETWORK_COMPRESS            = ON or OFF
OT_DBLOCK_CONTROL              = 0 or 1 (DBLOCK2 only)
OT_PRERELASE                   = A0201 (Undocumented
                                  feature)
```

2) Dataset Level:

```
OT_TRIM_WHITESPACE             = ON or OFF (NULL)
OT_SERIALGET_ORDERBY           = ON or OFF
```

3) DBFIND Mode 88: ITEM is not used, ARGUMENT contains the standard CONDITION in the WHERE clause, which must be SQL compliant, its column name and syntax must be in ORACLE query form. This mode supports TEXT only values.

Syntax of TEXT condition:

a) Operator **must be surrounded by BLANKS and must be in upper-case**, for example, COLUMN1 = ‘Lee’ AND COLUMN2 <> ‘TSAI’
b) DBFIND(88) is a very restricted in terms of its syntax. Supported operators are =,>,<,>=,<=,<>,LIKE, and OR,AND.

Only =, >, <, >=, <=, <>, LIKE
Only AND, OR
Must be in UPPER CASE
Must be surrounded by BLANKS for all operators
No special chars are allowed in STRING ‘ ‘ values

4) DBFIND Mode 89: similar to Mode 88, but handles BINARY values in Argument for the WHERE CONDITION clause.

New performance options in OTDRV:

5) –h for OPTIONS(DIRECT=TRUE) and UNRECOVERABLE for sqlldr.

6) –z for allowing NULL in VARCHAR2 NON-IMAGE-KEY and NON-IMAGE-SORT_FIELD columns, which must be used in conjunction with OT_TRIM_WHITESPACE = ON option.