OPENTURBO™ for DB2

iMaxsoft Corporation

Version B01.00

WE'LL ALWAYS BE HERE

Industry leader in database migration and middleware products
Based in Cupertino, CA since 1987

Revision History

Date	Revision	Author	Changes
6/7/2006	1.0	Oliver Wai	Version 1.0
9/24/2007	2.0	Victor Tsai	Version 2.0

Table of Contents

CONVENTIONS USED IN THIS MANUAL 4

OPENTURBO OVERVIEW 4

OPENTURBO LIBRARIES 5

OPENTURBO CONFIG FILE 7

CRITICAL RULES BEFORE YOU START 7
DATABASE LEVEL CONFIGURATION 8
DATASET LEVEL CONFIGURATION 11
ENCRYPTING/DECRYPTING PASSWORDS 12
SAMPLE CONFIG FILE 12

OPENTURBO SYNCHRONIZATION MODES 13

STARTING THE OPENTURBO LISTENER 13
FROM HPE3000 TO HP-UX 15
FROM HP-UX TO HPE3000 16

OPENTURBO UTILITIES 18

CROSSREF FILE 18
LANUTIL 23
LISTNER 24
OTGENLOADDB2 27
OTINTEGRITYDB2 28
RECVGEN, RECVPGEN 32
TIDRVA02DBG, TIDRVDB2 33
TIGENSECDB2 41
TILOAD, TILOADAM 42

DEBUGGING CAPABILITIES 44

APPENDIX 46

APPENDIX A: TROUBLESHOOTING 46
APPENDIX B: OPENTURBO ERROR MESSAGES 47
APPENDIX C: TURBOIMAGE DATA CONVERSION REFERENCE 52

Conventions Used In This Manual

- **Green Entries**: need modification when synchronizing from HP 9000 to HP e3000
- **Blue Entries**: specify the differences between [1] database to be synchronized and [2] database to be updated locally
- **Highlighted entries :** highlight special specification required
- **Red Entries**: error messages to pay attention to

OPENTURBO Overview

The OPENTURBO allows you to run your IMAGE legacy application with a Relational Database Management System (RDBMS) without any migration recoding. OPENTURBO is customizable and adaptable to a variety of infrastructure setup. With OPENTURBO it is now possible to run legacy applications on:

1. Running legacy applications on a HP 3000 server with a remote RDBMS backend system.
2. Running legacy applications on the RDBMS server (i.e. HP 9000)
3. Running legacy applications on a separate (non-HP 3000) application server with a RDBMS server.

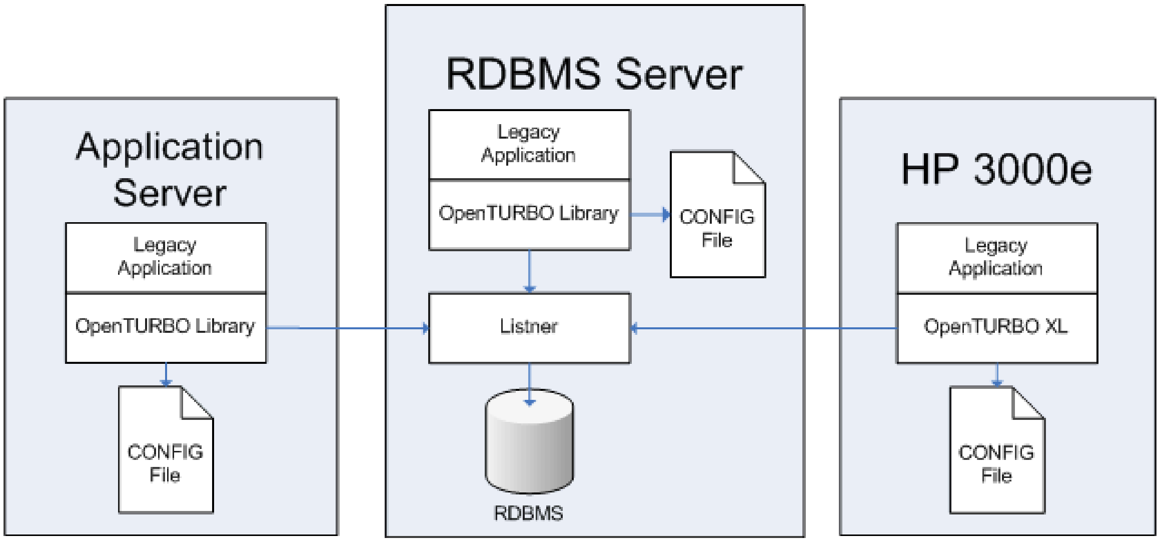

Figure 1: OPENTURBO Configurations

Alternatively, depending on the migration strategy, OPENTURBO gives the ability to maintain mirror database on a RDBMS and IMAGE DB. Known as, *DUALMODE*, OPENTURBO allows for unidirectional data replication and synchronization across an IMAGE database and any other RDBMS.

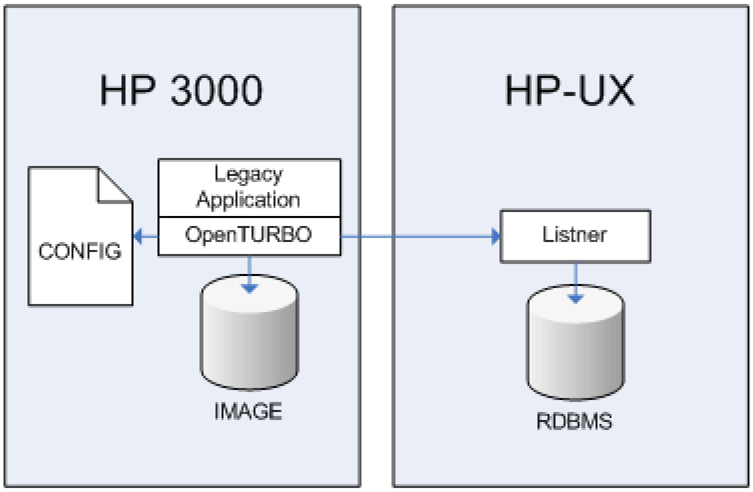

Figure 2: Mirror IMAGE & RDBMS w/legacy application on HP3000

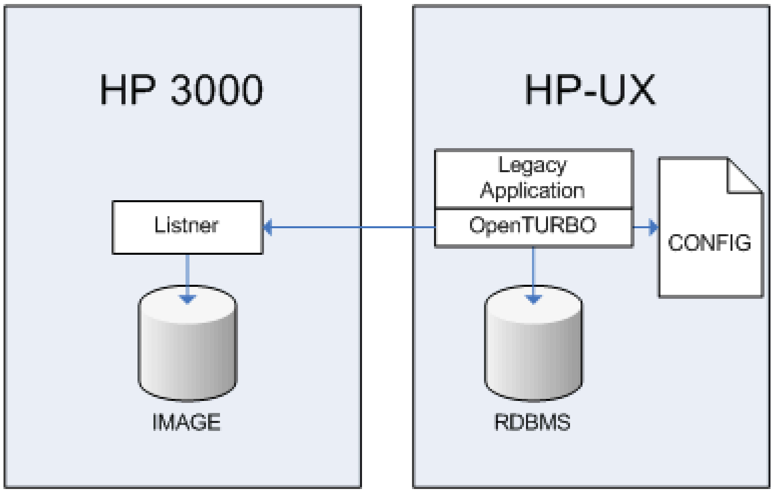

Figure 3: Mirror IMAGE & RDBMS w/ legacy application on HP-UX

The level of customization will allow enterprises flexibility in planning migration strategies, from simply migrating to a RDBMS backend to a full migration of both the business application and the database backend.

OPENTURBO Libraries

The OPENTURBO DB2 library is supported on HP-UX PA-RISC 32-bit architecture. The library components include:

Library	**Description**
`libdbutx.sl`	Library used for cleanup and security
`liblt.sl`	Middleware library
`libotDB2.sl`	Main API library
`libsdkDB2.sl`	Underlying server side database engine library

`libsdkc.sl`	Underlying client side database engine library
`libti.sl`	Library used to access OPENTURBO root file
`libtiam.sl`	Same as libti but with added Amysis and CROSSREF support

Dynamic client libraries on HPe3000 MPE/XL:

Library Name	Description
`LTXL`	AIM Middleware and Debugging Facility Core Library
`OTXL`	OPENTURBO Core Library
`OTXLDBG`	OPENTURBO Core Library with Debugger
`OTQRY`	OPENTURBO Core Library for QUERY.PUB.SYS
`OTQRYDBG`	OPENTURBO Core Library for QUERY.PUB.SYS with Debugger
`DRIVER`	Utilities Core Library
`TIDRV`	TurboIMAGE Test Driver Core Library

OPENTURBO CONFIG File

At the core of the OPENTURBO suite is the CONFIG file. The CONFIG File controls and manages the entire OPENTURBO Computing Environment. It allows you to customize replication parameters such as OS type, RDBMS type, database server, access methods, and update options. It is the most critical file and should be centrally and securely managed. Any improper changes to the CONFIG File can potentially damage the integrity of your databases and change your applications behaviors.

Critical Rules Before You Start

1. Preset OPENTURBO configuration files to the systems updating a centralized server. The OPENTURBO `CONFIG` file must be present on **ALL** systems using OPENTURBO. For example, if your environment uses a number of application servers that update a centralized database server, a copy of the `CONFIG` file must be present on each application server running `OPENTURBO.`

   You MUST use environment variable `OT_CONFIG` to redirect the `CONFIG` file name. For example on most UNIX environments:

   ```
   export OT_CONFIG=/pub/db/conf1
   ```

   On HPe3000 systems, there are two functionally equivalent ways of setting the CONFIG file name:

   ```
   # Set the OT_CONFIG variable
   :setvar OT_CONFIG ‘conf1’

   # Or Alternatively alias ‘conf1’ to the ‘config’ file name.
   : file CONFIG=conf1
   ```

2. All databases that the legacy application uses (including the ones will not to be synchronized) must be configured in the configuration file. These configurations include:
   a. Details about the local [and remote] databases
   b. If a database is to act as a mirror and must be synchronized
   c. If an application calls an undefined database through the OPENTURBO library, an error of **Error: “DBOPEN error -11”** will occur

3. If you are planning to do reverse data synchronization (from the RDBMS to HPe3000 where the TurboIMAGE resides, the following variables need to be specified in the configuration file: `TI_DUALMODE_HOST, TI_DUALMODE_SERVICE, TI_DUALMODE_PGM.`

4. Environment variables or the passing parameters cannot be used in the configuration files.

5. The CONFIG file must be a non-numbered text file.

6. Comments must start at the beginning of a line. For example:

```
// VALID COMMENT FORMAT
OT_WRITE_CACHE = WRITE_CACHE_LOCATION  // INVALID COMMENTS
```

Database Level Configuration

Field **Description**

OT_TI_DBNAME

Contains the fully qualified TurboIMAGE database name. OT_TI_DBNAME must match the database name stored in the source (or local) database. In addition, it must be set outside of a database definition block and set before database name alias. For example:

```
OT_TI_DBNAME = DBNAME.GROUP.ACCOUNT
```

You can have as many aliases as you want, but avoid making the same alias for different TurboIMAGE databases in the same CONFIG file. In all cases, the first found alias for a particular DB is used.

OT_ROOT_FILE

Points to the TIFILE root-file created by TILOAD. No environment variable is allowed in the file name. For example, if a name $HOME/db/DBNAMEti.ti is entered an error: 'TIFile cannot be found' is returned.

OT_RESERVE_WORD_FILE

Points to the file that contains relational database reserve words and conversion suffix. Default for HP 3000 is RESERVE.DB2.IMAXSOFT. For Unix servers, the default is:

```
/opt/imaxsoft/OPENTURBOx.x/db2/conf/RESERVE.DB2
```

OT_ERROR_FILE

Points to the file that contains the TurboIMAGE errors and messages for OPENTURBO. Default for HP3000 OTERROR.DB2.IMAXSOFT. For Unix servers, the default is:

```
/opt/imaxsoft/OPENTURBOx.x/db2/conf/OTERROR.DB2
```

OT_HOST

Points to the application server or the machine where your DB2 (or RDBMS) server is located. It is used when doing synchronization from HPe3000 to HP-UX. Field ignored when synchronization from the HP-UX to HP3000e direction.

OT_SERVICE

Directs OPENTURBO to the appropriate listener daemon on the OT_HOST. It is used when doing synchronization from HPe3000 to HP-UX. Field ignored when synchronization from the

HP-UX to HP3000e direction.

`OT_DB_RDBMS`

Hex combination of OS and RDBMS. For example, ELOQUENCE on HP-UX takes the HEX value of 0x0200 (DB_TYPE_HPUX) and adds it to 0x0009 (DB_TYPE_ELOQUENCE) resulting in 0x0209.

Note that Eloquence = ImageDB on non HP3000 platforms

Operating System:

```
   #define DB_TYPE_MPEXL            0x0100
   #define DB_TYPE_HPUX             0x0200 /* all UNIX
*/
   #define DB_TYPE_MSNT             0x0300 /* all
INTEL */
```

Database:

```
   #define DB_TYPE_ORACLE           0x0003
   #define DB_TYPE_SQLSVR           0x0004
   #define DB_TYPE_ELOQUENCE        0x0009
   #define DB_TYPE_IMAGE            0x000a
   #define DB_TYPE_DB2              0x0010
```

Some common combinations:

```
   ORACLE on HP-UX: 0x0203 (or 515 in decimal form)
   ORACLE on INTEL-LINUX: 0x0303 (or 772)
   SQL_SERVER on Windows: 0x0304 (or 771)
   ELOQUENCE on HP-UX: 0x0209 (or 521)
   DB2 on HP-UX : 0x210 (or 528)
```

`OT_RDB_LOGON`

The IMAGE database has a table of 64 customizable access roles. By setting the `OT_RDB_LOGON` to an IMAGE role, OPENTURBO will create a single general access role based on the access control of the given role. The format of this field is:

```
   <fully qualified name>/<encrypted password>
```

`OT_SDK_SERVER_PGM`

Contains the program name of OPENTURBO Core Server Program. By default the OPENTURBO uses:

```
   /opt/imaxsoft/OPENTURBOx.x/db2/lbin/dbsvr
```

This field only applies to OPENTURBO Client-Server mode.

`OT_CIUPDATE`

In IMAGE, keys update (applies only to detail dataset) is not allowed unless CIUPDATE is specified. `Set OT_CIUPDATE = ON` to allow critical item updates.

IF you use CIUPDATE, you also have to set the same option on iMaxSoft's DOOR (Data Object Open Replication) product that OPENTURBO uses to replicate databases:

1. Use TILOAD to generate map file TIFILE for main data structure.
2. Unload detail dataset and load into SQL Server with option –c (OTDRV).
3. Generate map file with option –c (DRMAPGEN).
4. Start shooter with option –c (SHOOTOT).

`OT_IMAGEMODE`

If `OT_IMAGEMODE=ON`, then only the IMAGE DB is updated. For example, if you run your applications on HP3000, then only the local IMAGE DB is updated. If are running your applications from HP9000, then it uses `TI_DUALMODE_HOST`, `TI_DUALMODE_SERVICE` and `TI_DUALMODE_PRG` to access IMAGE on the remote HP3000.

`OT_DUALMODE`

Different behaviors are set depending on the combination of `OT_DUALMODE` and `OT_IMAGEMODE`:

OS	IMAGEMODE	DUALMODE	Comment
MPE/XL	ON	OFF	TurboIMAGE only
MPE/XL	-	ON	DB2/others RDBMS primary TurboIMAGE secondary
MPE/XL	OFF	OFF	DB2/others RDBMS only
HP-UX	ON	OFF	TurboIMAGE only
HP-UX	-	ON	TurboIMAGE primary DB2/others RDBMS secondary
HP-UX	OFF	ON	DB2/others RDBMS primary TurboIMAGE secondary
HP-UX	OFF	OFF	DB2/others RDBMS only

For unidirectional data replication and synchronization across an IMAGE database and DB2 (see figure 2 & 3 above) set `OT_DUALMODE` to '2PC'. If replication is from HP-UX to HPe3000, then `TI_DUALMODE_HOST`, `TI_DUALMODE_SERVICE`, `TI_DUALMODE_PRG` variables must also be set.

`TI_DUALMODE_HOST,`
`TI_DUALMODE_SERVICE,`
`TI_DUALMODE_PGM`

Used to connect to TurboIMAGE on HP/3000 from your HP-UX applications. They are used ONLY used during synchronization from HP-UX to HP e3000. The fields are ignored (and can be omitted) when synchronizing from HP e3000 to HP-UX 9000.

NOTE: You must start the Listener `JLISTNER.PUB.IMAXSOFT` *from the HP/3000 before running your OPENTURBO* *applications and utilities on HP-UX in DUAL-MODE or OT_IMAGEMODE.*

`OT_TRX_THRESHOLD`

Reports performance bottlenecks where a SQL transaction executing time exceeds `OT_TRX_THRESHOLD` limit. By default the threshold is 10 seconds. OPENTURBO logs any execution that exceeds threshold limit and reports it in performance analysis.

Your programs must linked with `OTXLDBG` (HP e3000) or `libotdbg` (HP-UX) libraries, and OPENTURBO debugging facility must also be set either '`SETVAR LTDBG7 1`' or '`EXPORT LTDBG7=1`'.

For most applications, setting `OT_TRX_TRHESHOLD=2` is sufficient for detecting potential performance bottlenecks.

`OT_WRITE_CACHE`

DB2 uses the `OT_WRITE_CACHE` only when `DUALMODE=2PC`. OPENTURBO will synchronize both an OT cache buffer and the primary database on the local and remote systems.

There are 3 modes used for synchronization

`STATEMENT_COMMIT`

Data is updated to (local) primary database and cache buffer in an ATOMIC fashion (2PC), then the data is subsequently sent to remote side asynchronously *after each statement (INSERT, DELETE, UPDATE, etc)***.** OPENTURBO will perform sync-point check when the CACHE BLOCK (currently set to 30K) is full.

`OFF`

Waits for COMMIT Acknowledgement.

`OT_READ_CACHE,`
`OT_RDB_OWNER,`
`OT_LOCKWAIT_CYCLE,`
`OT_NETWORK_COMPRESS,`
`OT_LOCKCOVERAGE,`
`OT_DBLOCK_CONTROL`

N/A for DB2

Dataset Level Configuration

Field	**Description**
`OT_NOOPT`	Set to `ON` since none of the Dataset Level configurations are applicable for DB2.

ENCRYPTING/DECRYPTING Passwords

Passwords must be encrypted when using OPENTURBO. OPENTURBO provides a pair of functions to be used for passwords.

```
Syntax ENCRYPT.BIN.IMAXSOFT <decrypted password>
Example:    :ENCRYPT.BIN.IMAXSOFT WRITER
            OPENTURBO ENCRYPT <A.01.00> iMaxsoft Corp. Copyright 2002.
            Encrypted Password = [RMDOZM]

Syntax DECRYPT.BIN.IMAXSOFT <encrypted password>
Example:    :DECRYPT.BIN.IMAXSOFT RMDOZM
            OPENTURBO DECRYPT <A.01.00> iMaxsoft Corp. Copyright 2002.
            Original Password = [WRITER]
```

Sample CONFIG File

```
//    Denotes the comment lines
{}    Begin and End of a Database Definition Block
[]    Begin and End of a Dataset Definition Block
```

```
      OT_TI_DBNAME = INVENT.DATA.MOULTON
      INVENT.DATA.MOULTON {
      INVENT.DATA {
      INVENT {
         OT_ROOT_FILE         =
   /opt/imaxsoft/OPENTURBO3.7db2/db/inventti.ti
         OT_RESERVE_WORD_FILE =
   /opt/imaxsoft/OPENTURBO3.7db2/conf/RESERVE.DB2
         OT_ERROR_FILE        =
   /opt/imaxsoft/OPENTURBO3.7db2/conf/OTERROR.DB2
         OT_HOST              = 127.0.0.1
         OT_SERVICE           = 32601
         OT_OS_RDBMS          = 528
         OT_RDB_LOGON         = OT.db2inst2/gzz0122
         OT_SDK_SERVER_PGM    =
   /opt/imaxsoft/OPENTURBO3.7/db2/lbin/dbsvrDB2
         OT_RDB_OWNER         = INVENT_DATA_MOULTON
         OT_CIUPDATE          = ON
         OT_DUALMODE          = OFF
         TI_DUALMODE_HOST     = 207.92.64.65
         TI_DUALMODE_SERVICE  = 32602
         TI_DUALMODE_PGM      = DMDRV.BIN.IMAXSOFT
         OT_LOCKWAIT_CYCLE    = 60
         OT_TRX_THRESHOLD     = 1
      }
```

OPENTURBO Synchronization Modes

Synchronization by default consists of 2 data sources, one on local server and one on remote server. In order to access remote server you must start listener on the remote server. The database level configuration OT_DUALMODE must be set to'2PC' for unidirectional data replication and synchronization across an IMAGE database and DB2 database (see figure 2 & 3 above). If replication is from HP9000 to HPe3000, then TI_DUALMODE_HOST, TI_DUALMODE_SERVICE, TI_DUALMODE_PRG variables must also be set.

Starting the OPENTURBO Listener

You must start this daemon process on your HP9000 regardless if you access the database locally or remotely. The listener program accepts DBOPEN requests from your application programs, and then spawns the OPENTURBO server process DBSVR, which performs all subsequent database access calls. The listener is also responsible for OPENTURBO recovery; if DBSVR aborts abnormally, the listener will make sure all dangling database objects that are created by the DBSVR are clean-up properly.

You must provide an unused server port for listener to use, check /etc/services file and find an open number: the range is from 1 through 32768. It is highly recommended that you add the newly assigned entry into /etc/services file for ease of control.

Here is an example entry:

```
       OTB    32608/tcp    otb      # For OPENTURBO Listener
```

In the client, the listener connection control data is stored in the CONFIG file, OT_HOST, OT_SERVICE, OT_OS_RDBMS, OT_RDB_LOGON. OT_SDK_SERVER_PRG are used to connect to the target host machine, to talk to the listener, to spawn the server program, and to connect to the proper database via proper database logon.

Note: the OT_RDB_LOGON is used only when your program login is as the creator of the TurboIMAGE and use semicolon as the password. Otherwise, the DBOPEN password is mapped to its corresponding DB user.

Sample script to start a listener without a configuration file:

```
       export LTDBG17=0
       export LTDBG18=0
       export LTDBG19=0
       export LTDBG27=0
       export LTDBG28=0
       export LTDBGOUT=-
       /opt/imaxsoft/OPENTURBO3.7/db2/bin/listner 32601
```

You need to start the listener using the proper DB and OPENTURBO setup credentials, which means all environment variables, access paths, library paths, etc. must point to the proper location of your database, OPENTURBO programs, MF-COBOL, and dynamic libraries.

Do not turn on OPENTURBO debugging from listener level, but turn it on through `DBCONTROL`; if you turn on OPENTURBO debugging at listener level, the `LTDBGOUT` file will logs all clients' info. There is no way to isolate individual client trace. This feature is used only in the development environment; you can assign each programmer a listener, then he or she controls his/her own environment. One client per listener; this is the easy way to turn on OPENTURBO trace.

Start listener with configuration file:

```
$/opt/imaxsoft/OPENTURBO3.7/db2/bin/listner ^conf32601
```

Configuration conf32601 file format:

```
############################################################################
#  Copyright (c) iMaxsoft Corp. 2006           All Rights Reserved.
#
#   DEBUG MASK EXAMPLES:
#      0  3  4  7  8  11 12 15 16 19 20 23 24 27 28 31
#      0000  0000  0000  0000  0000  0000  0000  0000
#
#      APP1  40000000  1
#      APP2  20000000  2
#      LAN   00004000 17
#      SOCK  00002000 18
#      NIPC  00001000 19
#      SQLX2 00000010 27
#      SQLX1 00000008 28
#
############################################################################
[ GLOBAL ]
   SERVICE = 32601
   HOME = /tmp
   DBGOUT = /tmp/ltdbgout32601
#  DBGMASK = 60003000 APP1 + APP2 + NIPC + SOCK
   DBGMASK = 00000000
   SERVER = 207.0.0.1
   PARM =
   STANDBY = Q1
   NICE = /opt/imaxsoft/OPENTURBO3.7/db2/config/mynice

[ Q1 ]
   SERVER = /opt/imaxsoft/OPENTURBO3.7/db2/bin/mypgm1
   MIN = 2
   MAX = 10
```

The configuration has two parts: the GLOBAL and the QUEUE for standby processes.

Global Area

`SERVICE` Port of the listener

`HOME` Listener's home directory; this where core dump is placed for your application on HP-UX

`DBGOUT` Listener’s debug output file name

`DBGMASK` 32-bit mask, from left to right, each bit controls one level of debug trace. The leftmost bit is mapped to LTDBG1, and so on. FFFFFFFF turns all on and 00000000 turns all off.

`PARM` SERVER: Default is 127.0.0.1.
Mimics HP3000 MPE/XL run command’s parm= option

`STANDBY` list of standby queue names that are defined in the QUEUE block below; you can specify multiple names here separated by comma, i.e. Q1, Q2, Q3

`NICE` Specifies the file name that contains a list of programs that need to be spawned at different nice value. Unless the listener has SU capability, all nice value must be equal or less than listener’s. If nice failed due to lack of capability, then the spawned process is set to the same value as the listener.

The file format is:

```
/opt/imaxsoft/OPENTURBO3.7/db2/myprog1 30
/opt/imaxsoft/OPENTURBO3.7/db2/myprog2 19
/opt/imaxsoft/OPENTURBO3.7/db2/myprog3 20
/opt/imaxsoft/OPENTURBO3.7/db2/myprog4 -10
```

Queue Area

`SERVER` Name of the standby program
`MIN` Minimum number of standby programs to be started when listener is initiated
`MAX` number of standby programs that are allowed

From HPe3000 to HP-UX

1. Set the FILE equation to CONFIG file. For example:

```
FILE CONFIG=MYCONFIG.GROUP.ACCT
```

2. Start the Listener on the HP-UX.

3. Set values for CONFIG files.

```
      OT_TI_DBNAME = DBNAME.GROUP.ACCOUNT
      DBNAME.GROUP.ACCOUNT {
      DBNAME.GROUP {
      DBNAME {
        OT_IMAGEMODE          = OFF
        OT_ROOT_FILE          = DBNAMEti.ti
```

```
  OT_RESERVE_WORD_FILE = RESERVE.DB2
  OT_ERROR_FILE        = OTERROR.DB2
  OT_HOST              = 10.1.1.8
  OT_SERVICE           = 32600
  OT_OS_RDBMS          = 521
  OT_RDB_LOGON         = DBNAME_GROUP_ACCOUNT/YWIVHZ
  OT_SDK_SERVER_PGM    = /opt/imaxsoft/OPENTURBOx.x/db2/bin/dmdrv
  OT_CIUPDATE          = ON
  OT_DUALMODE          = 2PC
  OT_TRX_THRESHOLD     = 2
  OT_WRITE_CACHE       = BULK_COMMIT
  OT_READ_CACHE        = ON
OT_DETAILSETNAME = @ [
  OT_NOOPT = ON
]
}
```

From HP-UX to HPe3000

1. Make sure DB2 and OPENTURBO is set in the $SHLIB_PATH library path

2. Set the OT_CONFIG variable to point to the CONFIG

3. Start the Listener on the HPe3000

4. Set values for the CONFIG files.

```
OT_TI_DBNAME = DBNAME.GROUP.ACCOUNT
DBNAME.GROUP.ACCOUNT {
DBNAME.GROUP {
DBNAME {
  OT_IMAGEMODE         = OFF
  OT_ROOT_FILE         = DBNAMEti.ti
  OT_RESERVE_WORD_FILE = RESERVE.DB2
  OT_ERROR_FILE        = OTERROR. DB2
  OT_HOST              = 10.1.1.8
  OT_SERVICE           = 32600
  OT_OS_RDBMS          = 521
  OT_RDB_LOGON         = DBNAME_GROUP_ACCOUNT/YWIVHZ
  OT_SDK_SERVER_PGM    = /opt/imaxsoft/OPENTURBOx.x/db2/bin/dmdrv
  OT_CIUPDATE          = ON
  OT_DUALMODE          = 2PC
  TI_DUALMODE_HOST     = 10.1.1.66
  TI_DUALMODE_SERVICE  = 32602
  TI_DUALMODE_PGM      = DMDRV.BIN.IMAXSOFT
  OT_TRX_THRESHOLD     = 2
  OT_WRITE_CACHE       = BULK_COMMIT
  OT_READ_CACHE        = ON
OT_DETAILSETNAME = @ [
  OT_NOOPT = ON
]
}
```

5. Set IMAGE passwords (for access to SUPRTOOL/Query3k). Since the `listner` on the `HP3000` is not in the same group.acct as the IMAGE database, it is not possible to access the IMAGE DB without logging in. This can be done with the following code snippet:

```
export EQ_DBPASSWORD=USER
export EQ_DBUSER=MYPASSWORD
export EQ3K_<dbname>=<dbname>.<group>.<account>
```

OPENTURBO Utilities

CROSSREF File

The CROSSREF File is the Cross Reference File used to:

1. Change column name
2. Change type default override
3. Rename of default OPENTURBO internal key column name
4. re-position OPENTURBO internal key columns.

In addition to these core features, CROSSREF also allows you to exclude records based on DATA SEARCH PATTERN exclusion rules and to convert fields from NULLs to BLANK or BLANKS to NULL.

Special Notes

iMaxsoft has the option of adding additional columns known IMAXSOFT13_PATH_nn and IMAXSOFT13_SEQ_NO in migrated database to help preserve the IMAGE linked-list internal data structure. If needed (though not recommended), these fields can be used emulated the sequence in the IMAGE database if the sequence or traversal of data is important to your application. The prefix of IMAXSOFT13 was chosen as the default name for this field to avoid name conflicts but it is possible to choose a custom prefix by setting UKEY_NAME and PATH_PREFIX in the CROSSREF file.

CROSSREF Syntax

The CROSSREF syntax consists of 3 columns:

Action	**ITEM_NAME**	**Change To**
Action to be performed	The original IMAGE column name.	Value or column type that ITEM_NAME will be converted to.

The values of ITEM_NAME are in IMAGE format where the *usage of underscore is **NOT** allowed*. For example TOTAL-AMT is valid whereas TOTAL_AMT not.

Types of CROSSREF actions that can be performed include:

Action	**Description**
COLNAME_CHANGE	Replaces ITEM_NAME strings that match the pattern specified with the ChangeTo string. Rules of ITEM_NAME patterns are: 1) If iMaxsoft encounters a RESERVED word for column name, it appends a default suffix in order to make the name legal. COLNAME_CHANGE allows you to append a user-defined suffix for

RESERVED words.

2) ‘@’ character is used as a wildcard. @ is allowed at beginning or at end but not in the middle of the string.

For example, @STRING, STRING@, and @STRING@ are valid search strings but STR@ING is invalid.

3) You may concatenate dataset name DATASET.ITEM_NAME, then the name change applies to the specific dataset only

NOTE:
CROSSREF actions are order specific. It is important to put the more restricted rule first in order to achieve the intended result.

COLTYPE_CHANGE

Changes OPENTURBO default data type. Valid changes include:

X_TO_BINARY: VARCHAR to BINARY
X_TO_NUMBER: VARCHAR to NUMBER
CHAR: VARCHAR to CHAR

NOTE:
1. Length cannot be changed.
2. CROSSREF is order specific. If two commands change the same field, then the latter change overwrites the previous change.

OWNER

OPENTURBO will use this user defined owner name as the sole owner for all TurboIMAGE databases referencing this CROSSREF file. The ITEM_NAME is unused in this action and is reserved for future use. The string ‘NA’ must be entered in the ITEM_NAME column.

NOTE
The same owner name must be properly set in the run-time CONFIG file for database access.

UKEY_NAME

Unique key name, default name is IMAXSOFT13_SEQ_NO. Setting this field will allow user-defined name for this column.

PATH_PREFIX

Path chronological order column name prefix default name is IMAXSOFT13. Setting this field will allow user-defined name for this column.

UKEY_ORDER

re-position OPENTURBO internal key column order

FIRST_COL - put internal key columns at beginning

Sample CROSSREF Entries

Action	ITEM_NAME	Change To	Description
COLTYPE_CHANGE	TOTAL-AMT	X_TO_BINARY	Change TOTAL-AMT column type from VARCHAR to binary
COLNAME_CHANGE	@#@	_NBR	Replace all columns with '#' with '_NBR' in column name
COLTYPE_CHANGE	YMD@	X_TO_NUMBER	Change all columns with the pattern YMD@ from VARCHAR to number
COLTYPE_CHANGE	@	CHAR	Change all columns from VARCHAR to CHAR
UKEY_NAME	IMAXSOFT13_SEQ_NO	IMAGE_RECNBR	Changes column name to IMAGE_RECNBR
PATH_PREFIX	MAXSOFT13	IMAGE	Changes column name to IMAGE
UKEY_ORDER	IMAXSOFT13_SEQ_NO	FIRST_COL	Puts internal key columns as the first column in migrated table.
OWNER	NA	NEWOWNER	Change owner to AMIOWN. The ITEM_NAME is not used and must be set to 'NA'.

HP3000 Sample Script

You need to use **`TILOADAM`** to generating your `TIFILE`, since `TILOAD` doesn't support `–e` option.

```
setvar ltdbg1 0
setvar ltdbg2 0
setvar ltdbg3 0
setvar ltdbg4 0
setvar ltdbg6 0
setvar ltdbgout '$stdlist'
echo
echo 1) use tiloadam, not tiload for TIFile creation, tiload doesn't
echo    support -e CROSSREF file
echo 2) CROSSREF file is crossref.config
echo 3) Must use crossref.config CROSSREF for both tiloadam and otdrv60
echo
```

```
input NAME=ot_go;prompt="OK to proceed?"
echo
echo ---- create inventti.ti
echo
tiloadam.bin.ims0100;info='-d invent.data.sampledb -t inventti.ti &
-r reserve.db2 -e crossref.config'
echo
echo ---- unload and direct load invent18 (SI dataset)
echo
otdrv60.bin.ims0100;info=' &
-dinvent.data.sampledb -t20 &
-ti inventti.ti &
-v reserve.db2 &
-s si &
-recnum -b DB2 -e crossref.config &
-g 207.92.64.9:32688:INVENT:INVENT:/opt/imaxsoft/db2/otldr2:&
2000:/tmp/'
```

Use `TILOADAM` to generate your `TIFILE` instead of `TILOAD. TILOADAM` allows you to specify a CROSSREF file using the `–e option.` The proper loading script and data file will be created according to your CROSSREF using otdrv. Sample run result:

```
LT928A: MGR.IMS0100(81): gobdir

1) use tiloadam, not tiload for TIFile creation, tiload doesn't
   support -e CROSSREF file
2) CROSSREF file is crossref.config
3) Must use crossref.config CROSSREF for both tiloadam and otdrv60

OK to proceed?

---- create inventti.ti


OPENTURBO TILOAD <A.01.04> iMaxsoft Corp. Copyright 2002.

IMAXSOFT/CSF IMAXSOFT Corp. Copyright 2002-2004, All Rights Reserved.
License No. 000000                       DEMO                       [2005/08/05]


Process Start: 2005-07-16 09:28:50
Process Stop:  2005-07-16 09:28:55

---- unload and direct load invent18 (SI dataset)


OPENTURBO*Pro OTDRVEZ <A.02.01> iMaxsoft Corp. Copyright 2003.

OPENTURBO IMAXSOFT Corp. Copyright 2002-2004, All Rights Reserved.
License No. 000000                       DEMO                       [2005/08/05]


OPENTURBO TIUNLOAD Dataset    [SI]
          UNLOAD Data File    [invent18.OTDATA.IMS0100]
          UNLOAD Script File  [invent18.OTSCRIPT.IMS0100]
          COPY of the Dataset [invent18.OTCOPY.IMS0100]
          WORKING File        [invent18.OTWORK.IMS0100]

OPENTURBO OTDRV Parms:
          -dinvent.data.sampledb
          -tiinventti.ti
          -vreserve.db2
          -t20
          -rinvent18.OTDATA.IMS0100
          -linvent18.OTSCRIPT.IMS0100
          -oinvent18.OTCOPY.IMS0100
          -winvent18.OTWORK.IMS0100
          -sSI
          -bDB2
          -ecrossref.config
           EXCEPTION REPORT = invent18.OTEXCEPT.IMS0100
           OTEDIT_OFF
           CHRONOLOGICAL_OFF
           ASCII_TRANSFER
           NON_EXCLUSIVE_MODE
           USE_TI_RECNUM and USE_FOR_DOOR
           DB2
           NORMAL_LOAD_SCRIPT
           KEEP_WHITESPACE
           GO_STRAIGHT_TO_ORACLE
              ORACLE Host     =207.92.64.9
              OPENTURBO Port  =32688
              ORACLE User     =INVENT
              ORACLE Password =INVENT
              OPENTURBO Loader=/opt/imaxsoft/db2/otldr2
              LOGFILE dir     =/tmp/
              COMMIT Count    =2000

DB2:  TOTLen         = 4076
DB2:  RDBFileRecSize = 4076
DB2:  MEDIARecSize   = 508

RDBFileRecSize  = 4076
MEDIARecSize    = 1016
OTGenDetailOutFile: Set(18) HW(124487:1) Row(47690) Reject(0)

Process Start:  2005-07-16 09:28:58
Process Stop:   2005-07-16 10:01:22
Process Elapse: 00:32:24
LT928A: MGR.IMS0100(82):
```

lanutil

Lanutil allows you to view any process that is currently connected to your DB2 database locally or remotely through OPENTURBO. This utility can be run on the net, as long as you specify the host name and service port number; it directs you all the way to the specific listener and reports the status.

Lanutil is also used to gracefully shut down the listener, command: STOPALL.

```
   LANUTIL (A.06.00.00) iMAXSOFT Corp. Copyright 1993-2002, All
Rights Reserved.


   HOST:[127.0.0.1]  APPLICATION:[32601]

   Commands: LIST    - shows all connected users.
             KILL id - kills the specified user.
             STOPALL - terminates listener and all users.
             HOST id - sets to new host node name.
             APPL id - sets to new application name.
             SETQ qname #servers
                     - sets # of standby servers for a queue
             EXIT    - ends the LANUTIL.
   LANUTIL>>
```

1. LIST command, shows all server processes that are spawned by the listener running on the HOST [127.0.0.1] and waiting on the SERVICE [32601].
2. KILL id command, kills the specific server process via the ID from the LIST command report.
3. STOPALL, is the best way to gracefully shut down this listener along with associated child processes.
4. HOST id command, reconnects Lanutil to another HOST via either an IP address or a DNS name.
5. APPL id command, reconnects Lanutil to another listener via either a SERVICE number or a SERVICE name.
6. SETQ qname command, OPENTURBO listener supports persistent and parallel stand-by modes. At current release of OPENTURBO, only persistent listener is supported; ignore this command.
7. EXIT command, ends Lanutil program.

listner

The listner daemon process must be started on your HP9000 regardless if you access your database locally or remotely. The listener program accepts `DBOPEN` requests from your application programs, and then spawns the OPENTURBO server process `DBSVR`, which performs all subsequent database access calls. The listener is also responsible for OPENTURBO recovery; if DBSVR aborts abnormally, the listener will make sure all dangling database objects that are created by the DBSVR are clean-up properly.

You must provide an unused server port for listener to use, check `/etc/services` file and find an open number: the range is from 1 through 32768. It is highly recommended that you add the newly assigned entry into `/etc/services` file for ease of control. Here is an example entry:

```
       OTB    32608/tcp    otb      # For OPENTURBO Listener
```

In the client, the listener connection control data is stored in the CONFIG file, `OT_HOST`, `OT_SERVICE`, `OT_OS_RDBMS`, `OT_RDB_LOGON`. `OT_SDK_SERVER_PRG` are used to connect to the target host machine, to talk to the listener, to spawn the server program, and to connect to the proper database via proper database logon.

Note: the OT_RDB_LOGON is used only when your program login is as the creator of the TurboIMAGE and use semicolon as the password. Otherwise, the DBOPEN password is mapped to its corresponding database user.

Sample script to start a listener without a configuration file:

```
       export LTDBG17=0
       export LTDBG18=0
       export LTDBG19=0
       export LTDBG27=0
       export LTDBG28=0
       export LTDBGOUT=-
       $ /opt/imaxsoft/OPENTURBO3.7/db2/bin/listner 32601
```

You need to start the listener with super user capability and from the login with proper DB2 and OPENTURBO setup, which means all environment variables, access paths, library paths, etc. must point to the proper places for DB2 database and OPENTURBO, MF-COBOL, DB2 dynamic libraries.

Do not turn on OPENTURBO debugging from listener level, but turn it on through `DBCONTROL`; if you turn on OPENTURBO debugging at listener level, the `LTDBGOUT` file will logs all clients' info. There is no way to isolate individual client trace. This feature is used only in the development environment; you can assign each programmer a listener, then he or she controls his/her own environment. One client per listener; this is the easy way to turn on OPENTURBO trace.

HP-UX Special Features

1. If you name your listner to listner*ND*, then the listener will turns itself into a Non Daemon mode.

2. The client process can pass program name with options. For example:

- If `;shlib=/imaxsoft:/imaxsoft/a` is part of your program name, then `/imaxsoft/pub:/imaxsoft/a` will be inserted by listener to the front of `$SHLIB_PATH`

- If `;cwd=/imaxsoft/tmp` is part of your program name, then listener will change your program home directory to `/imaxsoft/tmp`.

listner Configuration File

The configuration has two parts: the GLOBAL and the QUEUE for standby processes.

Global Definition Area

`SERVICE` Port of the listener

`HOME` Listener's home directory; this where core dump is placed for your application on HP-UX

`DBGOUT` Listener's debug output file name

`DBGMASK` A 32-bit mask, from left to right, each bit controls one level of debug trace. The leftmost bit is mapped to LTDBG1, and so on. FFFFFFFF turns all on and 00000000 turns all off.

`SERVER` Always loop-back 127.0.0.1 or local host name

`PARM` Mimics HP3000 MPE/XL run command's parm= option

`STANDBY` List of standby queue names that are defined in the QUEUE block below; you can specify multiple names here separated by comma, i.e. Q1, Q2, Q3

`NICE` Specify the file name that contains a list of programs that need to be spawned at different nice value. Unless the listener has SU capability, all nice value must be equal or less than listener's. If nice failed due to lack of capability, then the spawned process is set to the same value as the listener.

The file format is:

```
/opt/imaxsoft/OPENTURBO3.7/db2/myprog1 30
/opt/imaxsoft/OPENTURBO3.7/db2/myprog2 19
/opt/imaxsoft/OPENTURBO3.7/db2/myprog3 20
/opt/imaxsoft/OPENTURBO3.7/db2/myprog4 -10
```

Queue Definition Area

`SERVER` Name of the standby program

`MIN` Minimum number of standby programs to be started when listener is initiated

`MAX` Maximum number of standby programs that is allowed

Sample Configuration File

```
###############################################################################
#  Copyright (c) iMaxsoft Corp. 2006           All Rights Reserved.
#
#   DEBUG MASK EXAMPLES:
#      0  3  4  7  8  11 12 15 16 19 20 23 24 27 28 31
#      0000  0000  0000  0000  0000  0000  0000  0000
#
#      APP1  40000000  1
#      APP2  20000000  2
#      LAN   00004000 17
#      SOCK  00002000 18
#      NIPC  00001000 19
#      SQLX2 00000010 27
#      SQLX1 00000008 28
#
###############################################################################
[ GLOBAL ]
   SERVICE = 32601
   HOME = /tmp
   DBGOUT = /tmp/ltdbgout32601
#  DBGMASK = 60003000 APP1 + APP2 + NIPC + SOCK
   DBGMASK = 00000000
   SERVER = 207.0.0.1
   PARM =
   STANDBY = Q1
   NICE = /opt/imaxsoft/OPENTURBO3.7/db2/config/mynice

[ Q1 ]
   SERVER = /opt/imaxsoft/OPENTURBO3.7/db2/bin/mypgm1
   MIN = 2
   MAX = 10
```

otgenLOADDB2

otgenLOADDB2 converts table load script file created into DB2 Command Line Interface script.

```
OPENTURBO OTGenLoad <A.01.00> iMaxsoft Corp. Copyright 2002.

usage: otgenLOAD -u -d -t -c -x
       -d  TurboIMAGE Fully Qualified Database Name
           DBName.GROUP.ACCOUNT
       -t  A TABLE_NAME or @ for ALL_TABLES of the
           specified TurboIMAGE database
       -c  CACHE (default) or NOCACHE
           CACHE   is to get multiple sequence
                   numbers per request
           NOCACHE is to get single sequence
                   number at a time
           Use NOCACHE in all cases, unless there is a
           performance degradation
       -x  Loader Script Output Filename
           Default = OTLoadDB.EXECUTE - executes all following
                                     scripts
                     OTLoadDB.DROP    - drops all table objects for
                                        performance
                     OTLoadDB.SQLLDR  - loads data into ORACLE
                                        database
                     OTLoadDB.GEN     - generates all table objects
                                        after successful loading
```

Example

```
/opt/imaxsoft/OPENTURBO3.7/db2/bin/otgenLOAD -u ot/JO                 \
                         -d MUSIC.TIDATA.IMAXSOFT \
                         -t @                     \
                         -c NOCACHE
```

The resultant output script file `OTLoadDB.EXECUTE` will contain the following script:

```
sqlplus ot/ot @OTLoadDB.DROP
OTLoadDB.SQLLDR
sqlplus ot/ot @OTLoadDB.GEN
```

`OTLoadDB.DROP` drops objects for speedy loading. `OTLoadDB.SQLLDR` loads data into corresponding ORACLE table. `OTLoadDB.GEN` then recreates all table objects after the loading.

otINTEGRITYDB2

otINTEGRITY scans and checks DB2 database objects created by OPENTURBO at TurboIMAGE level to ensure the integrity of the DB2 database and TIFile. otINTEGRITY reports any integrity related discrepancies and errors, you must shutdown the database and fix all errors to prevent any further damages to your DB2 database. All passwords must be entered in their encrypted form.

```
OPENTURBO otINTEGRITY <A.01.04> iMaxsoft Corp. Copyright 2002

usage: otINTEGRITY -u -d -t -r -p
       -u Database Login User/Encrypted Password
       -d TurboIMAGE Fully Qualified Database Name
          DBName.GROUP.ACCOUNT.
       -t TI File Name
       -r Database RESERVE WORD File Name
       -p Database Encrypted Password.for User
          DBName_GROUP_ACCOUNT.
```

otINTEGRITY will check for:

Database Structure

- Table: table name, table type (logical type, A, M, and D), and table columns which includes column name, column position, column type, column length and NULL indicator.
- Index: index name, index type (unique, non-unique, cluster, etc.) and key columns which includes column name and column position.
- Constraints: primary keys and foreign keys, which is mainly used for maintaining A and M to D relationships.

Database Unique Number

- Sequence: compare the maximum value of all OPENTURBO internal control columns IMAXSOFT13_PATH_nn and IMAXSOFT13_SEQ_NO to the next_val of table sequence object and to ensure there is no unique constrain violation.

Database Triggers

- Trigger: trigger name, trigger type (insert, update or delete), and trigger contents.

Database Securities

- Authority: access authorities for OPENTURBO control tables DBACCESS, DBLOCK, DBLOCK1, DBLOCK2, and DOOR_SYNC. DB2 users and passwords (TurboIMAGE classes and passwords), and DB2 user's database objects access authorities, objects are tables, columns, sequences, synonym (name domain), session, DBA, and DDL.

Database Checksums

- otCHECKChecksum - verifies checksums in DB2 OT.TIBASE and OPENTURBO TIFile.

Examples

```
   $_OTB_BIN/otINTEGRITY -u OT/JO \
                         -d MUSIC.TIDATA.IMAXSOFT \
                         -r /opt/imaxsoft/OPENTURBO3.7/db2/conf/RESERVE.DB2
\
                         -t
/opt/imaxsoft/OPENTURBO3.7/db2/musicdemo/tidb/ti \
                         -p HPNDX > output_music
```

The corresponding output for this database will return:

```
OPENTURBO otINTEGRITY <A.01.04> iMaxsoft Corp. Copyright 2002

DB2_USER      - [OT/JO]
TIBASE_NAME      - [MUSIC.TIDATA.IMAXSOFT] [MUSIC_TIDATA_IMAXSOFT]
TIFile           - [/opt/imaxsoft/OPENTURBO3.7/db2/musicdemo/tidb/ti]
RWFile           - [/opt/imaxsoft/OPENTURBO3.7/db2/conf/RESERVE.DB2]
OWNER_PASSWD     - [HPNDX]

----- DB2 TABLES of TurboIMAGE[MUSIC.TIDATA.IMAXSOFT]:
DBIntegrity - [ALBUMS                  ] OT_SETS_NUM=[0]
DBIntegrity - [COMPOSERS               ] OT_SETS_NUM=[1]
DBIntegrity - [DBACCESS                ] OT_SETS_NUM=[2]
DBIntegrity - [DBLOCK                  ] OT_SETS_NUM=[3]
DBIntegrity - [DBLOCK1                 ] OT_SETS_NUM=[4]
DBIntegrity - [DBLOCK2                 ] OT_SETS_NUM=[5]
DBIntegrity - [DOOR_SYNC               ] OT_SETS_NUM=[6]
DBIntegrity - [LOG                     ] OT_SETS_NUM=[7]
DBIntegrity - [SELECTIONS              ] OT_SETS_NUM=[8]
DBIntegrity - [SELECTIONS_A            ] OT_SETS_NUM=[9]

----- DB2 TABLE Count = [10], TurboIMAGE Set Count = [5]

***** TurboIMAGE Dataset=[ALBUMS][M ][1] Table=[ALBUMS]:
OT_GetSetInfo - [ALBUMCODE               ] COLUMNSCnt=[0]
OT_GetSetInfo - [ALBUMTITLE              ] COLUMNSCnt=[1]
OT_GetSetInfo - [MEDIUM                  ] COLUMNSCnt=[2]
OT_GetSetInfo - [ALBUMCOST               ] COLUMNSCnt=[3]
OT_GetSetInfo - [RECORDINGCO             ] COLUMNSCnt=[4]
OT_GetSetInfo - [DATERECORDED            ] COLUMNSCnt=[5]
OT_GetSetInfo - [MFGCODE                 ] COLUMNSCnt=[6]
OT_GetSetInfo - [COMMENT_IMS             ] COLUMNSCnt=[7]
OT_GetSetInfo - [IMAXSOFT13_SEQ_NO       ] COLUMNSCnt=[8]

===== TABLE[ALBUMS] Column Summary:
WARN [OT_GetSetInfo] Column[IMAXSOFT13_SEQ_NO       ] has no name match.

===== SEQUENCE=S01_ALBUMS CurrValue=18

***** TurboIMAGE Dataset=[COMPOSERS][M ][2] Table=[COMPOSERS]:
OT_GetSetInfo - [COMPOSERNAME            ] COLUMNSCnt=[0]
OT_GetSetInfo - [BIRTH                   ] COLUMNSCnt=[1]
OT_GetSetInfo - [DEATH                   ] COLUMNSCnt=[2]
```

```
           OT_GetSetInfo - [BIRTHPLACE              ] COLUMNSCnt=[3]
           OT_GetSetInfo - [COMMENT_IMS             ] COLUMNSCnt=[4]
           OT_GetSetInfo - [IMAXSOFT13_SEQ_NO       ] COLUMNSCnt=[5]

           ===== TABLE[COMPOSERS] Column Summary:
           WARN [OT_GetSetInfo] Column[IMAXSOFT13_SEQ_NO       ] has no name match.

           ===== SEQUENCE=S01_COMPOSERS CurrValue=11

           ***** TurboIMAGE Dataset=[SELECTIONS-A][A ][3] Table=[SELECTIONS_A]:
           OT_GetSetInfo - [SELECTIONNAME           ] COLUMNSCnt=[0]
           OT_GetSetInfo - [IMAXSOFT13_SEQ_NO       ] COLUMNSCnt=[1]

           ===== TABLE[SELECTIONS_A] Column Summary:
           WARN [OT_GetSetInfo] Column[IMAXSOFT13_SEQ_NO       ] has no name match.

           ===== SEQUENCE=S01_SELECTIONS_A CurrValue=14

           ***** TurboIMAGE Dataset=[SELECTIONS][D ][4] Table=[SELECTIONS]:
           OT_GetSetInfo - [ALBUMCODE               ] COLUMNSCnt=[0]
           OT_GetSetInfo - [SELECTIONNAME           ] COLUMNSCnt=[1]
           OT_GetSetInfo - [COMPOSERNAME            ] COLUMNSCnt=[2]
           OT_GetSetInfo - [TIMING                  ] COLUMNSCnt=[3]
           OT_GetSetInfo - [PERFORMERS              ] COLUMNSCnt=[4]
           OT_GetSetInfo - [COMMENT_IMS             ] COLUMNSCnt=[5]
           OT_GetSetInfo - [IMAXSOFT13_PATH_01      ] COLUMNSCnt=[6]
           OT_GetSetInfo - [IMAXSOFT13_PATH_02      ] COLUMNSCnt=[7]
           OT_GetSetInfo - [IMAXSOFT13_PATH_03      ] COLUMNSCnt=[8]
           OT_GetSetInfo - [IMAXSOFT13_SEQ_NO       ] COLUMNSCnt=[9]

           ===== TABLE[SELECTIONS] Column Summary:
           WARN [OT_GetSetInfo] Column[IMAXSOFT13_PATH_01      ] has no name match.
           WARN [OT_GetSetInfo] Column[IMAXSOFT13_PATH_02      ] has no name match.
           WARN [OT_GetSetInfo] Column[IMAXSOFT13_PATH_03      ] has no name match.
           WARN [OT_GetSetInfo] Column[IMAXSOFT13_SEQ_NO       ] has no name match.

           ===== SEQUENCE=S01_SELECTIONS CurrValue=12

           ***** TurboIMAGE Dataset=[LOG][D ][5] Table=[LOG]:
           OT_GetSetInfo - [ALBUMCODE               ] COLUMNSCnt=[0]
           OT_GetSetInfo - [SELECTIONNAME           ] COLUMNSCnt=[1]
           OT_GetSetInfo - [STARTTIME               ] COLUMNSCnt=[2]
           OT_GetSetInfo - [ENDTIME                 ] COLUMNSCnt=[3]
           OT_GetSetInfo - [ANNOUNCER               ] COLUMNSCnt=[4]
           OT_GetSetInfo - [IMAXSOFT13_PATH_01      ] COLUMNSCnt=[5]
           OT_GetSetInfo - [IMAXSOFT13_PATH_02      ] COLUMNSCnt=[6]
           OT_GetSetInfo - [IMAXSOFT13_SEQ_NO       ] COLUMNSCnt=[7]

           ===== TABLE[LOG] Column Summary:
           WARN [OT_GetSetInfo] Column[IMAXSOFT13_PATH_01      ] has no name match.
           WARN [OT_GetSetInfo] Column[IMAXSOFT13_PATH_02      ] has no name match.
           WARN [OT_GetSetInfo] Column[IMAXSOFT13_SEQ_NO       ] has no name match.

           ===== SEQUENCE=S01_LOG CurrValue=12

           ----- Database Integrity Checking Summary:
           WARN [DBIntegrity] Table[DBACCESS                ] has no name match.
           WARN [DBIntegrity] Table[DBLOCK                  ] has no name match.
           WARN [DBIntegrity] Table[DBLOCK1                 ] has no name match.
           WARN [DBIntegrity] Table[DBLOCK2                 ] has no name match.
           WARN [DBIntegrity] Table[DOOR_SYNC               ] has no name match.

           ----- Checksum Verification:

           << otCHECKChecksum DB2_USER = [OT/JO] >>
           << otCHECKChecksum TIBASE_NAME = [MUSIC.TIDATA.IMAXSOFT] >>
           << otCHECKChecksum TIFile      =
        [/opt/imaxsoft/OPENTURBO3.7/db2/musicdemo/tidb/ti] >>

           Checksum Summary Report:

           ----------- ------- FROM DATABASE -------- --------- FROM TIFile --------
           DBNAME    : MUSIC.TIDATA.IMAXSOFT          MUSIC.TIDATA.IMAXSOFT
           TIFILE    : ../tidb/ti
        /opt/imaxsoft/OPENTURBO3.7/db2/musicdemo/tidb/ti
           VERSION   : A.01.04                        A.01.04
           ITEM-CKSUM: 171                            171
           DSET-CKSUM: 305                            305
           PATH-CKSUM: 96                             96

           otCHECKChecksum Result = 0
                                1 = Mactched
                                0 = Not Mactched
                               -1 = Error

           Process Start: 2002-11-11 16:58:30
           Process Stop:  2002-11-11 16:58:37
```

recvgen, recvpgen

Recvgen generates sqlplus script from OPENTURBO cache file that can be used for recovery of DB2 database

```
      OPENTURBO recvgen<A.03.07> iMaxsoft Corp. Copyright 2006

      usage: recvgen RDBMS LogfileName OutfileName
             RDBMS Type  - ORACLE or DB2
             LogfileName - * for all files from current dir
             OutfileName – SQL Statements script output file
```

Similarly recvpgen is used as a verification version of recvgen. It is used to pre-generate SQLStatement from a cache file that can be checked for accuracy.

```
      OPENTURBO recvpgen<A.03.07> iMaxsoft Corp. Copyright 2006

      usage: recvpgen RDBMS LogfileName
             RDBMS Type  - ORACLE or DB2
             LogfileName - * for all files from current dir
```

tidrvA02dbg, tidrvDB2

TIDRV is OPENTURBO testing driver program. TIDRV is the best tool used to verify data migration results, to perform progressive test, and to conduct performance benchmarking analysis. You can enter TurboIMAGE API calls using its straight-forward syntax and verify the formatted results.

TIDRV can be run in silent mode, which takes inputs from a command file and reports output to an output file or in interactive mode which uses SDTIN and STDOUT as the input and output files.

TIDRV can be run in OPENTURBO mode as well as in TurboIMAGE mode. In OPENTURBO mode, it accesses DB2 database either locally or remotely. In TurboIMAGE mode, it can be configured to access TurboIMAGE database locally or remotely. The result formats are identical in both modes and you can 'diff' them easily.

Running TIDRV

```
OPENTURBO TIDRV <A.01.04> iMaxsoft Corp. Copyright 2002.

run tidrv.bin;info='-r -i -o -map -turboimage'
   -rRWFile    OPENTURBO Reserve Word File Name
   -iINPUT     TurboIMAGE Command Input File Name
   -oOUTPUT    Output File Name
   -map        Print TurboIMAGE Schema
   -turboimage Access TurboIMAGE Directly via OPENTURBO Library
```

Option **Description**

-r OPENTURBO Reserve Word File Name. You may use absolute file `$/opt/imaxsoft/OPENTURBO3.7/db2/conf/RESERVE.ORACLE` directly, or copy `RESERVE.ORACLE` into your login MPE account, or use MPE file equation command `:FILE RESERVE.ORACLE=` to locate the file, or use -r to identify the `RESERVE.ORACLE` file.

-map Prints TurboIMAGE schema

-i Specifies the input command file name

- HP3000: use $stdin for interactive mode
- HP9000: use - for interactive mode

-o Specifies the output result file name

- HP3000: Use $stdlist for terminal output
- HP9000: Use - for terminal output.

On HP e3000

- Running `TIDRV;XL=”OTXL.A.IMAXSOFT”` accesses remote DB2 database on HP9000.
- Running `TIDRV;XL=”XL.PUB.SYS”` accesses local TurboIMAGE database.
- Running `TIDRV;XL=”OTXL.A.IMAXSOFT”;INFO=”-turboimage”` accesses local TurboIMAGE database via OPENTURBO MPE/XL library

On HP9000

- Running `TIDRV` with `libot.sl` accesses local DB2 database
- Running `TIDRV -turboimage` with `libot.sl` accesses remote TurboIMAGE database on HP e3000 via OPENTURBO HP-UX library

Dual Mode

It is possible to run DUAL-MODE from both HP e3000 and HP9000.

TIDRV Rules and Syntax

- `Use // or /* for comment line`
- `Use COMMENT ON and COMMENT OFF for comment block`
- `Use & at end of each command line for denoting command continuation`
- `All value must be embedded in single quote ‘’`
- `Use \ for de-reference special character, such as \`
- `Use , for parameters separator`
- `PRINT ON and PRINT OFF to turn on and off print-result-to-file option`
- `DEFINE is used to declare variables, currently we only support short and int, which are 16-bit and 32-bit interger`
- `REPEAT n, executes the immediate followed DBCall n times, one call only`
- `DEBUGOUT filename, specifies the remote debugging file name (on HP9000)`
- `DEBUGn ON and DEBUGn OFF to turn on and off debugging level from 0 through 31; currently supported levels are:`

```
DEBUG0:     Serious Error (no need to turn on)
DEBUG1:     OPENTURBO core level trace
DEBUG2:     OPENTURBO reserved word
DEBUG3:     OPENTURBO mapped error message (TurboIMAGE)
DEBUG4:     OPENTURBO emulator level trace
DEBUG5:     OPENTURBO client SQL statement and CURSOR POOL trace
DEBUG6:     OPENTURBO DULA MODE diff results
DEBUG7:     OPENTURBO transaction performance trace
DEBUG13:    OPENTURBO call pattern analyzer
DEBUG17:    Network traffic dump in hex and text
DEBUG18:    Network socket trace
```

DEBUG19:	Network Net/IPC and TCP/IP trace
DEBUG27:	SQL statement dump
DEBUG28:	SQL error analyzer
DEBUG29:	malloc() and free() trace

- LOADTI,ti-filename and UNLOADTI (obsolete); loads the OPENTURBO root-file into memory for DBCalls syntax and semantic checking

- USETI,tifile-id; sets the active OPENTURBO root-file (tifile) for subsequent DBCalls syntax and semantic checking; the first LOADTI gets 0 for tifile-id, the second LOADTI gets 1 for tifile-id, and so on till 63

- EXIT ends TIDRV program

TIDRV TurboIMAGE Calls

1) DBOPEN , TurboIMAGE_Database_Name; , Password; , Mode

   - The first DBOPEN’s baseID = 0
   - The second DBOPEN’s baseID = 1
   - The third DBOPEN’s baseID = 2
   - . . . and so on

2) DBCLOSE , baseID , Dataset-Name | Dataset-Number | None , Mode

   - The baseID is the number associated to the DBOPEN
   - None means nothing in between two commas, such as ,,
   - The Dataset-Name is 16 characters long or terminated by either blank or semi-colon, such as MEMBERDETL;
   - The Dataset-Number is number only, such as 24

3) DBFIND , baseID , Dataset-Name | Dataset-Number , Mode , ITEM=‘Item-Name | Item-Number’ , ARG=Defined-Variable | ’Value’

   - The ITEM= is TIDRV’s key word and is part of comamnd syntax
   - The Item-Name is the key item, it can be 16 characters long or terminated by either blank or semi-colon, such as MBRNO
   - The Item-Number is number only, such as 5
   - The ARG= is TIDRV’s key word and is part of command syntax
   - The Defined-Variable is declared via TIDRV DEFINE command
   - The Value can be a true value, value with wildcard, or the standard argument as specified in TurboIMAGE manual page 180
   - OPENTURBO version A.01.00 and above support all modes but 10, which has been implemented but has not been certified by TPI vendors yet.

4) DBGET , baseID , Dataset-Name | Dataset-Number , Mode , LIST=‘Item-Name List | Item-Number List | Special List’ , ARG=Defined-Variable | ’Value’

   - The LIST= is TIDRV’s key word and is part of command syntax
   - Item-Name List is a list of item names separated by comma, such as MBRNO, MBRNAME, . . .
   - Item-Number List is a list of item numbers separated by comma, such as 24, 5, . . .
   - Special List has specific meaning, such as @; means all items, *; means same as previous DBCall List, and so on
   - ARG= is used for Manual Master calculated get by key value or direct get by record number

5) DBERROR is part of DBEXPLAIN, use DBEXPLAIN instead

6) DBEXPLAIN , baseID

7) DBCONTROL , baseID , QUALIFIER=’’ , Mode

- The QUALIFIER= is TIDRV’s key word and is part of command syntax
- Supported modes:

Mode 5: Enables the critical item update option
Mode 6: Disables the critical item update option
Mode 7: Allows Dynamic Multiple Database Transaction
Mode 88: Turns ON/OFF a remote debugger level, use the first half-word of QUALIFIER= for the debugging level (0 through 31) and the second half-word of QUALIFIER= for the ON(1) and OFF(0) switch
Mode 89: Sets the remote debugger file name, such as QUALIFIER=’debugger file name’

8) DBINFO , baseID , QUALIFIER=’’ , Mode

- Refer to TurboIMAGE manual for QUALIFIER=, page 190

9) DBLOCK , baseID , QUALIFIER=’’ , Mode

- Refer to TurboIMAGE manual for QUALIFIER=, page 207 shows the detail format of the lock descriptor
- CLIENT-LOCK-MANAGER is responsible for checking and enforcing process related lock rules
- SERVER-LOCK-MANAGER is responsible for checking and enforcing cross-process lock rules

10) DBUNLOCK , baseID , None , Mode

11) DBPUT , baseID , Dataset-Name | Dataset-Number , Mode , LIST=’’ , DATA=’’

12) DBDELETE , baseID , Dataset-Name | Dataset-Number , Mode

13) DBUPDATE , baseID , Dataset-Name | Dataset-Number , Mode , LIST=’’ , DATA=’’

14) DBXBEGIN , baseID | baseID:baseID:baseID:. . , Mode

15) DBXEND , baseID | baseID:baseID:baseID:. . , Mode

16) DBXUNDO , baseID | baseID:baseID:baseID:. . , Mode

Examples

```
// HP3000 Run script

    parm xl='NOOT'
    setvar xlflag '!xl'
    setvar ltdbg1 0
    setvar ltdbg2 0
    setvar ltdbg3 0
    setvar ltdbg4 1
    setvar ltdbg5 1
    setvar ltdbg6 1
    setvar ltdbg17 0
    setvar ltdbg18 0
    setvar ltdbg19 0
    setvar ltdbgout '$stdlist'
    purge outtrx1.output
    if (xlflag = 'OT') then
```

```
       file config=config.lee.ims
       run tidrv.bin;xl='otxldbg.a,tidrv.a,ltxl.pub.imaxsoft'; &
       info='-iintrx1.lee -oouttrx1.output'
    else
       run tidrv.bin;xl='tidrv.a,ltxl.pub.imaxsoft'; &
       info='-iintrx1.lee -oouttrx1.output'
    endif

// 1. OT triggers TIDRV to run in OPENTURBO emulator mode
// 2. NOOT triggers TIDRV to run in normal TurboIMAGE mode
// 3. The input command file name is intrx1
// 4. The output result file name is outtrx1
```

// HP3000 Input command file:

```
LOADTI,ti1.ti
// ** DBPUT to Detail Dataset - PURCHASE
DBOPEN,INVENT.DATA.MOULTON;,FAVOR;,3
// DBOPEN,INVENT.DATA.MOULTON;,;,1
//
COMMENT ON
DEBUGOUT /opt/imaxsoft/OPENTURBO3.7/db2tmp/lee.dbg
DEBUG19 ON
DEBUG18 ON
DEBUG17 ON
DEBUG27 ON
DEBUG28 ON
COMMENT OFF
//
DBCONTROL,0,,7
//
//         ==========>> Only for OPENTURBO <<===============
// ==>> TIDRV doesn't suppoty mutiple DBOPEN in TurboIMAGE Mode <<==
//
//*** DBXDBGIN Syntax = DBXBEGIN,BASE=id/BASELIST=id:id:id,Mode ***
//*** DBXEND   Syntax = DBXEND,BASE=id/BASELIST=id:id:id,Mode ***
//*** DBXUNDO  Syntax = DBXUNDO,BASE=id/BASELIST=id:id:id,Mode ***
//
DBXBEGIN,BASE=0,1
//
// Dataitem Level LOCK +++++ CC_PNLC_PO = X[36]
// ------------     EQUAL Confition    ------------------
DBLOCK,0,ARG='1,36,PURCHASE;,CC-PNLC-PO,=,&
CC_PNLC_PO_88                       ',5
DBPUT,0,PURCHASE;,1,LIST='@;',DATA='CC_PNLC_PO_88,POR_KEY_88,88,88,88,&
88,88,88,REV_LOT,REQ_NO,COMMENT_IMS,88,USER_IMS,20020526,888888'
DBUNLOCK,0,,1
// ------------ <= and >= No ERROR when DBPUT ------------------
DBLOCK,0,ARG='2,36,PURCHASE;,CC-PNLC-PO,<=,&
CC_PNLC_PO_95                       ,36,PURCHASE;,CC-PNLC-PO,>=,&
CC_PNLC_PO_70                       ',5
DBPUT,0,PURCHASE;,1,LIST='@;',DATA='CC_PNLC_PO_89,POR_KEY_89,89,89,89,&
89,89,89,REV_LOT,REQ_NO,COMMENT_IMS,89,USER_IMS,20020526,999999'
DBUNLOCK,0,,1
//
// Rewind Dataset PURCHASE
// ** Backward Serial DBGET 1 Records from PURCHASE and DBDELETE it
//
// Rewind Dataset PURCHASE and DBGET the Last 4 Records
//
DBCLOSE,0,PURCHASE,2
DBGET,0,PURCHASE;,3,LIST='@;',ARG=''
DBGET,0,PURCHASE;,3,LIST='@;',ARG=''
//
// DBXEND,BASE=0,1
DBXUNDO,BASE=0,1
//
DBCLOSE,0,,1
UNLOADTI
//
```

```
// HP3000 Output result file:

=>> Repeat[1] TICommand[LOADTI,ti1.ti]
=>> Repeat[1] TICommand[DBOPEN,INVENT.DATA.MOULTON;,FAVOR;,3]
DBOPEN,INVENT.DATA.MOULTON;,FAVOR;,3,status[1]=0,status[2]=1
=>> Repeat[1] TICommand[DBCONTROL,0,,7]
DBCONTROL: --------------------
mode=[7]
status[1]   =[0]
status[2]   =[1]
status[3-4] =[0]
status[5-6] =[0]
status[7-8] =[0]
status[9-10]=[0]
=>> Repeat[1] TICommand[DBXBEGIN,BASE=0,1]
DBXBEGIN: --------------------
mode=[1]
status[1]   =[0]
status[2]   =[1]
status[3-4] =[0]
status[5-6] =[0]
status[7-8] =[0]
status[9-10]=[0]
=>> Repeat[1] TICommand[DBLOCK,0,ARG='1,36,PURCHASE;,CC-PNLC-PO,=,CC_PNLC_PO_88
                      ',5]
DBLOCK: --------------------
base=[INVENT.DATA.MOULTON;]
mode=[5]
status[1]   =[0]
status[2]   =[1]
status[3]   =[0]
status[4]   =[0]
status[5-6] =[0]
status[7-8] =[0]
status[9-10]=[0]
=>> Repeat[1] TICommand[DBPUT,0,PURCHASE;,1,LIST='@;',DATA='CC_PNLC_PO_88,POR_KE
Y_88,88,88,88,88,88,88,REV_LOT,REQ_NO,COMMENT_IMS,88,USER_IMS,20020526,888888']
DBPUT: --------------------
base=[INVENT.DATA.MOULTON;]
dset=PURCHASE;[15]
mode=[1]
list=@;
DATA BUFFER Begin ====================
CC-PNLC-PO         1X36  =>CC_PNLC_PO_88
POR-KEY            1X20  =>POR_KEY_88
VEND-NO            1I2   =>+0000000088
QTY-ORD            1I2   =>+0000000088
DATE-ORD           1I2   =>+0000000088
DATE-PROM          1I2   =>+0000000088
DATE-LAST          1I2   =>+0000000088
REV-LOT            1X10  =>REV_LOT
REQ-NO             1X16  =>REQ_NO
COMMENT            1X30  =>COMMENT_IMS
UNIT-COST          1I4   =>+0000000000000000088
USER               1X8   =>USER_IMS
DATE-UPDT          1I2   =>+0020020526
TIME-UPDT          1I2   =>+0000888888
DATA BUFFER End   ====================
status[1]   =[0]
status[2]   =[80]
status[3-4] =[500]
status[5-6] =[0]
status[7-8] =[0]
status[9-10]=[0]
=>> Repeat[1] TICommand[DBUNLOCK,0,,1]
DBUNLOCK: --------------------
base=[INVENT.DATA.MOULTON;]
mode=[1]
status[1]   =[0]
status[2]   =[1]
status[3-4] =[500]
```

```
status[5-6] =[0]
status[7-8] =[0]
status[9-10]=[0]
=>> Repeat[1] TICommand[DBLOCK,0,ARG='2,36,PURCHASE;,CC-PNLC-PO,<=,CC_PNLC_PO_95
                       ,36,PURCHASE;,CC-PNLC-PO,>=,CC_PNLC_PO_70
       ',5]
DBLOCK: --------------------
base=[INVENT.DATA.MOULTON;]
mode=[5]
status[1]   =[0]
status[2]   =[1]
status[3]   =[0]
status[4]   =[500]
status[5-6] =[0]
status[7-8] =[0]
status[9-10]=[0]
=>> Repeat[1] TICommand[DBPUT,0,PURCHASE;,1,LIST='@;',DATA='CC_PNLC_PO_89,POR_KE
Y_89,89,89,89,89,89,89,REV_LOT,REQ_NO,COMMENT_IMS,89,USER_IMS,20020526,999999']
DBPUT: --------------------
base=[INVENT.DATA.MOULTON;]
dset=PURCHASE;[15]
mode=[1]
list=@;
DATA BUFFER Begin ====================
CC-PNLC-PO         1X36  =>CC_PNLC_PO_89
POR-KEY            1X20  =>POR_KEY_89
VEND-NO            1I2   =>+0000000089
(93/193) Continue?
QTY-ORD            1I2   =>+0000000089
DATE-ORD           1I2   =>+0000000089
DATE-PROM          1I2   =>+0000000089
QTY-RECD           1I2   =>+0000000089
DATE-LAST          1I2   =>+0000000089
REV-LOT            1X10  =>REV_LOT
REQ-NO             1X16  =>REQ_NO
COMMENT            1X30  =>COMMENT_IMS
UNIT-COST          1I4   =>+0000000000000000089
USER               1X8   =>USER_IMS
DATE-UPDT          1I2   =>+0020020526
TIME-UPDT          1I2   =>+0000999999
DATA BUFFER End   ====================
status[1]   =[0]
status[2]   =[80]
status[3-4] =[502]
status[5-6] =[0]
status[7-8] =[0]
status[9-10]=[0]
=>> Repeat[1] TICommand[DBUNLOCK,0,,1]
DBUNLOCK: --------------------
base=[INVENT.DATA.MOULTON;]
mode=[1]
status[1]   =[0]
status[2]   =[0]
status[3-4] =[502]
status[5-6] =[0]
status[7-8] =[0]
status[9-10]=[0]
=>> Repeat[1] TICommand[DBCLOSE,0,PURCHASE,2]
DBCLOSE,INVENT.DATA.MOULTON;[0],PURCHASE[15],2,db_status[1]=0
=>> Repeat[1] TICommand[DBGET,0,PURCHASE;,3,LIST='@;',ARG='']
DBGET: --------------------
base=[INVENT.DATA.MOULTON;]
dset=PURCHASE;[15]
mode=[3]
list=@;
DATA BUFFER Begin ====================
CC-PNLC-PO         1X36  =>CC_PNLC_PO_89
POR-KEY            1X20  =>POR_KEY_89
VEND-NO            1I2   =>+0000000089
QTY-ORD            1I2   =>+0000000089
DATE-ORD           1I2   =>+0000000089
```

```
DATE-PROM          1I2   =>+0000000089
QTY-RECD           1I2   =>+0000000089
DATE-LAST          1I2   =>+0000000089
REV-LOT            1X10  =>REV_LOT
REQ-NO             1X16  =>REQ_NO
COMMENT            1X30  =>COMMENT_IMS
UNIT-COST          1I4   =>+0000000000000000089
USER               1X8   =>USER_IMS
DATE-UPDT          1I2   =>+0020020526
TIME-UPDT          1I2   =>+0000999999
DATA BUFFER End   ====================
status[1]   =[0]
status[2]   =[80]
status[3-4] =[502]
status[5-6] =[0]
status[7-8] =[0]
status[9-10]=[0]
=>> Repeat[1] TICommand[DBGET,0,PURCHASE;,3,LIST='@;',ARG='']
DBGET: --------------------
base=[INVENT.DATA.MOULTON;]
dset=PURCHASE;[15]
mode=[3]
list=@;
DATA BUFFER Begin ====================
CC-PNLC-PO         1X36  =>CC_PNLC_PO_88
POR-KEY            1X20  =>POR_KEY_88
VEND-NO            1I2   =>+0000000088
QTY-ORD            1I2   =>+0000000088
DATE-ORD           1I2   =>+0000000088
DATE-PROM          1I2   =>+0000000088
QTY-RECD           1I2   =>+0000000088
DATE-LAST          1I2   =>+0000000088
REV-LOT            1X10  =>REV_LOT
REQ-NO             1X16  =>REQ_NO
COMMENT            1X30  =>COMMENT_IMS
UNIT-COST          1I4   =>+0000000000000000088
USER               1X8   =>USER_IMS
DATE-UPDT          1I2   =>+0020020526
TIME-UPDT          1I2   =>+0000888888
DATA BUFFER End   ====================
status[1]   =[0]
status[2]   =[80]
status[3-4] =[500]
status[5-6] =[0]
status[7-8] =[0]
status[9-10]=[0]
=>> Repeat[1] TICommand[DBXUNDO,BASE=0,1]
DBXUNDO: --------------------
mode=[1]
status[1]   =[0]
status[2]   =[80]
status[3-4] =[500]
status[5-6] =[0]
status[7-8] =[0]
status[9-10]=[0]
=>> Repeat[1] TICommand[DBCLOSE,0,,1]
DBCLOSE,INVENT.DATA.MOULTON;[8224],[-1],1,db_status[1]=0
=>> Repeat[1] TICommand[UNLOADTI]

// HP9000 Run script:

export LTDBGOUT=-
$_OTB_BIN/tidrv –iintrx1 –oouttrx1

// (Input command file: Same as HP3000)
// (Output result file: Same as HP3000)
```

tigenSECDB2

Reads OPENTURBO TI root-file and replicates TurboIMAGE security, which includes database passwords, data set and data item access classes, to the DB2 environment. All passwords must be entered in their encrypted form.

```
   OPENTURBO tigenSEC <A.01.00> iMaxsoft Corp. Copyright 2002.
   usage: tigenSEC -d -t -r -p -s -x
      -d     TurboIMAGE Fully Qualified Database Name
(DBName.GROUP.ACCOUNT).
      -t     OPENTURBO TI Rootfile
      -r     RESERVE WORD File
      -p     User DBName_GROUP_ACCOUNT Password
      -s     Default Tablespace for storing DB2 User
             Security Objects that are replicated from
             TurboIMAGE Access Class and Password
      -x     Output DB2 Security Script File Name

      *NOTE*
```

Notes

- You must have the OPENTURBO TI Rootfile (or a symbolic link to the rootfile) in the local directory to the TI Roofile. In OPENTURBO, the fully qualified TurboIMAGE database name is the TI Rootfile name.
- You must put the reserved word file in the directory `/opt/imaxsoft/OPENTURBO3.7/db2/conf` ***OR*** use -r to identify the location of the reserved word file.

Examples

```
   /opt/imaxsoft/OPENTURBO3.7/db2/bin/tigenSEC -d MUSIC.TIDATA.IMAXSOFT \
                           -t MUSIC.TIDATA.IMAXSOFT \
                           -r
/opt/imaxsoft/OPENTURBO3.7/db2/conf/RESERVE.DB2  \
                           -p HPNDX \
                           -s MUSIC
```

This command will create `OTCreateSEC.EXECUTE` script file that can be run to set up the security environment.

tiload, tiloadam

Generates the OPENTURBO TurboIMAGE (TI) File for the target database. tiloadam is identical to tiload, except that it *requires* a Cross Reference File.

Syntax

```
tiloadam.bin;info='-d -t -r -v -m -s -g –n -i'
tiloadam.bin;info='-d -t -r -v -m -s -g –n –i -e '
```

-dDBName	TurboIMAGE Database Name
-tTIFile	OPENTURBO Root-File Name
-rRWFile	OPENTURBO Reserve Word File Name. By default, TI looks for the file "Reserve" in current directory.
-v	OPTIONAL: The command will print on the terminal the TI version and checksum of the specified TIFile and the current IMAGE DB.
-mOUTFile	OPTIONAL: Prints OPENTURBO Root-File Structure to OUTFile
-sOUTFile	OPTIONAL: Generates TurboIMAGE Schema to OUTFile
-gSCHEMA	OPTIONAL: Generates OPENTURBO Root-File from a TurboIMAGE Schema File (SCHEMA). TurboIMAGE name is not allowed to qualify its group and account in the SCHEMA, so the -d DBName is used instead. This process performs TurboIMAGE schema syntax validation first and then creates TIFile.
-nItemList	OPTIONAL: Item List File contains a list of TurboIMAGE X-type Data Item Name(s) to be converted (i.e. X to K and mapped to RAW in ORACLE for NLS or X to double bytes in field)
-iFOSET	OPTIONAL: If set, OPENTURBO will display IMAGE dataset statistics from the TIFile FOSET on stdout.
-eCROSSREF	tiloadam ONLY: Cross Reference File for column name and type default override, rename of default OPENTURBO internal key column name and re-position OPENTURBO internal key columns. For DOOR: you may specify record exclusion rules which are DATA SEARCH PATTERNS, and rule to convert column's NULL byte to BLANK.

NOTES

1. Use –m option to get a TurboIMAGE and RDBMS mapping
2. Use –s option to regenerate your original TurboIMAGE database schema from an existing TIFILE.
3. Use –g option to recreate TIFILE directly from TurboIMAGE database schema.
4. Use –v to check version of the TIFILE
5. The database name is stored in the TIFILE for reference and internal use only. It is mainly used for our tool to cross-check the integrity of a TIFILE.
6. OWNER name is not stored since it is a run-time setting in RDBMS. Tables with the same name are differentiated by OWNER at runtime depending on the supplied UNIX login and schema name to the application using OPENTURBO.

Examples

1. Generating TI FILE

```
tiloadam.bin.imaxsoft;info='-dINVENT.DATAE.MOULTON -tINVENTTI  &
                              -rRESERVE.DB2.IMAX'
```

2. Comparing Version of TI File to IMAGE DB

```
tiloadam.bin.imaxsoft;info='-v -tINVENTTI -rRESERVE.DB2.IMAX'
```

3. Outputting IMAGE DB to File

```
tiloadam.bin.imaxsoft;info='-tINVENTTI –mOUT1 –rRESERVE.DB2.IMAX’
```

Debugging Capabilities

OPENTURBO supports 32 debugging levels starting from 0 through 31. Currently the following levels are defined:

```
LTDBG0       - OPENTURBO Internal Core ERROR
LTDBG1       - OPENTURBO Core Library Call Trace
LTDBG2       - OPENTURBO Reserved Words
LTDBG3       - OPENTURBO Error Messages
LTDBG4       - OPENTURBO Emulator Call Trace
LTDBG5       - OPENTURBO SDK Call Trace and CURSOR POOL Size
LTDBG6       - OPENTURBO DUAL MODE Diff Results
LTDBG7       - OPENTURBO Transaction Performance Trace
LTDBG13      - TurboIMAGE Call Flow Trace
LTDBG17      – Network Traffic Dump in Hex and Text formats
LTDBG18      – Socket Information
LTDBG19      – Net/IPC Information
LTDBG27      – Dynamic SQL Statement Preparation Trace
LTDBG28      – SQL Statement Execution Error
LTDBG29      – MALLOC, CALLOC and FREE Tracing
```

Example: Setup debugging from Server LISTENER Process.
The following setup will trigger all server DBSVR processes share the same debugging output file `/tmp/2002-07-16.dbg` with same debugging levels, 4, 17, 18, 19, 27, 28, and 29.

```
      export LTDBG4=1
      export LTDBG17=1
      export LTDBG18=1
      export LTDBG19=1
      export LTDBG27=1
      export LTDBG28=1
      export LTDBG29=1
      export LTDBGOUT= /tmp/2002-07-16.dbg
      ./listner OTB
```

Example: Setting up HP e3000 DUAL-MODE Differ Option to Validate Database migration.

Your can turn on internal DUAL-MODE option from OPENTURBO HP3000 emulator library by setting OT_DUALMODE = ON in the CONFIG file and you must also set the followings environment variables in order to view diff results:

```
      SETVAR LTDBG6 1
      SETVAR LTDBGOUT “difffile.group.account”
      RUN yourpgm
```

Example: Setting up HP-UX DUAL-MODE Differ Option to Validate Application migration.

DMDRV.PUB.IMAXSOFT is the HP3000 DUAL-MODE driver program that connects to your HP9000 programs for handling TurboIMAGE native APIs remotely.

On HP3000, you must stream the listener job first, JLISTNER.PUB.IMAXSOFT, which spawn child process DMDRV.PUB.IMAXSOFT to handle all remote TurboIMAGE calls from your HP9000 program.

JLISTNER File

```
       !job listener,mgr.imaxsoft
       !COMMENT
       !COMMENT **************************************************
       !COMMENT *  IMAXSOFT LISTENER - for DUAL-MODE from HPUX   *
       !COMMENT **************************************************
       !COMMENT
       !file hosts.net.sys=hosts.pub.imaxsoft
       !file services.net.sys=services.pub.imaxsoft
       !purge DMDIFF.LEE > $NULL
       !build DMDIFF.LEE;rec=-80,,f,ascii;disc=100000;msg
       !setvar ltdbg6 1
       !setvar ltdbgout "DMDIFF.LEE"
       !run listner.bin.imaxsoft;info="DBA"
       !eoj
```

On HP9000, you must set the followings in the CONFIG file:

```
       …
       OT_DUALMODE           = ON
       TI_DUALMODE_HOST      = 207.92.64.66
       TI_DUALMODE_SERVICE   = 32600
       TI_DUALMODE_PGM       = DMDRV.BIN.IMAXSOFT
       …
```

APPENDIX

Appendix A: Troubleshooting

Here is a list of some common issues:

1. Did you set the correct library path in your environment?

The correct library path must be set so that your application can find OPENTURBO libraries. For example:

```
export SHLIB_PATH=/opt/imaxsoft/OPENTURBO3.7/db2/lib
```

2. Did you specify an OT_ERROR_FILE?

A valid OT_ERROR_FILE must exist prior to synchronization.

3. Did you specify an OT_RESERVE_WORD_FILE?

A valid OT_RESERVE_WORD_FILE must exist prior to synchronization.

4. Did you start the listener on the remote server?

Communication on the remote server is handled by the listener. If the listener has not been started then there will be errors communicating with the remote server.

5. Do you have a valid license?

HP-UX

Run `/opt/leetech/bin/ltvalida`
Check product number 1006 and 1007

HPe3000

Run `OTVALIDA.PUB.IMAXSOFT`
Check product number 2006 and 1688

Appendix B: OPENTURBO Error Messages

```
       400:: GENERAL
       0     = SUCCESSFUL EXECUTION - NO ERROR
       -1    = NO SUCH DATABASE
       -11   = BAD DATABASE NAME OR PRECEDING BLANKS MISSING
       -12   = DATABASE MUST BE IN LOGON GROUP AND ACCOUNT
       -13   = NOT ALLOWED; MUST BE CREATOR OF ROOT FILE OR DATABASE
       -21   = BAD PASSWORD
       -22   = MAINTENANCE WORD REQUIRED
       -23   = USER (CLASS) LACKS WRITE ACCESS TO DATA SET
       -31   = DBGET MODE ILLEGAL  FOR DETAIL DATA SET
       -32   = UNOBTAINABLE ACCESS MODE
       -33   = MODE 7 DIAGNOSIS NOT ALLOWED
       -34   = DATABASE MUST BE RECOVERED BEFORE ACCESS IS ALLOWED
       -51   = LIST TOO LONG OR NOT PROPERLY TERMINATED
       -52   = ITEM SPECIFIED IS NOT AN ACCESSIBLE SEARCH ITEM IN THE SPECIFIED
    SET
       -53   = DBPUT LIST IS MISSING A SEARCH OR SORT ITEM
       -82   = CIUPDATE IS SET TO DISALLOWED; CANNOT USE CRITICAL ITEM UPDATE
       -90   = ROOT FILE BAD
       -99   = UNSUPPORTED FEATURE
       -121  = ILLEGAL LOCK DESCRIPTOR COUNT
       -123  = ILLEGAL RELATIONAL OPERATOR
       -124  = DESCRIPTOR LENGTH ERROR; MUST BE 9 OR MORE
       -125  = ILLEGAL SET NAME OR NUMBER IN DESCRIPTOR
       -126  = ILLEGAL ITEM NAME OR NUMBER IN DESCRIPTOR
       -127  = ILLEGAL ATTEMPT TO LOCK ON A COMPOUND ITEM
       -128  = VALUE FIELD TOO SHORT FOR THE ITEM SPECIFIED
       -129  = P28 IS LONGEST P-TYPE ITEM THAT CAN BE LOCKED
       -130  = ILLEGAL DECIMAL DIGIT IN TYPE 'P' DATA VALUE
       -131  = LOWERCASE CHARACTER IN TYPE 'U' DATA VALUE
       -132  = ILLEGAL DIGIT IN TYPE 'Z' DATA VALUE
       -133  = ILLEGAL SIGN CHARACTER IN TYPE 'Z' DATA VALUE
       -134  = TWO LOCK DESCRIPTORS CONFLICT IN SAME REQUEST
       -135  = DBLOCK CALLED WITH LOCKS ALREADY IN EFFECT IN THIS JOB/SESSION
       -136  = DESCRIPTOR LIST LENGTH EXCEEDS 4094 BYTES
       -137  = USER ABOUT To WAIT FOR SELF
       -139  = INVALID NUMBER OF BASE IDs
       -140  = BAD BASE ID LIST
       -151  = TEXT LENGTH GREATER THAN 512 BYTES
       -198  = TOTAL DBOPEN COUNT PER USER EXCEEDS LIMIT
       -212  = DATABASE CORRUPTION DETECTED
       -229  = CANNOT DELETE MANUAL MASTER WITH EMPTY CHAINS
       -258  = INVALID ARGUMENT FOR INDEX
       -259  = INVALID MODE FOR INDEX
       -260  = NO PREVIOUS LIST OF QUALIFIED DATA ENTRIES
       -305  = INVALID DATA SET NUMBER
       -306  = INVALID DATA SET TYPE
       -307  = INVALID RECORD NUMBER FOUND
       -420  = FEATURE NOT IMPLEMENTED
       -421  = BTE: UNKNOWN QUALIFIER VALUE FOR DBCONTROL MODE 13
       -422  = BTE: DATE SET # NOT IN VALID RANGE
       -423  = BTE: B-TREE ALREADY EXISTS
       -424  = BTE: FAILED TO CREATE B-TREE
       -425  = BTE: DB NOT OPENED EXCLUSIVELY
       -426  = BTE: B-TREE DOESN'T EXIST
       -429  = BTE: DBFIND ARGUMENT VERSION IS BAD
       -430  = BTE: DBFIND (mode 4/24) ARGUMENT TYPE IS BAD
       -431  = BTE: DBFIND (mode 4/24) ARGUMENT #1 LENGTH IS BAD
       -432  = BTE: WILDCARD NOT ASCII
       -433  = BTE: DBFIND (mode 4/24) ARGUMENT #2 LENGTH IS BAD
       -434  = DATASET DETAIL INSTEAD OF MASTER
       -436  = BTE: FAILED TO EXTRACT DATA FROM ROOT FILE
       -437  = BTE: FAILED TO CONVERT @c TO [] DBFIND
       -439  = BTE: CONVERSION OF KEY FROM EXTERNAL TO INTERNAL FORMAT FAILED
       -444  = BTE: DBFIND ON NON-KEY FAILED OF MASTER
       -446  = BTE: ARGUMENT 2 SPECIFIED FOR RELOP OF (</<=/=/>=/>)
       -452  = BTE: KEY LENGTH GREATER THAN 252 BYTES (MAXIMUM INDEX KEY SIZE)
       -458  = DBOPEN FAILED. OUT OF DISK SPACE
       10    = BEGINNING OF FILE
       11    = END OF FILE
       12    = DIRECTED BEGINNING OF FILE
       13    = DIRECTED END OF FILE
       14    = BEGINNING OF CHAIN
       15    = END OF CHAIN
       16    = THE DATA SET IS FULL
       17    = THERE IS NO CHAIN FOR THE SPECIFIED SEARCH ITEM VALUE
       18    = BROKEN CHAIN - FORWARD AND BACKWARD POINTERS NOT CONSISTENT
       20    = DATABASE CURRENTLY LOCKED SETS OR ENTRIES LOCKED WITHIN DATABASE
       22    = DATA SET ALREADY LOCKED
       23    = CANNOT LOCK SET DUE TO LOCKED ENTRIES WITHIN IT
       24    = ENTRIES CURRENTLY LOCKED USING DIFFERENT ITEM
       25    = CONFLICTING DATA ENTRY LOCK ALREADY IN EFFECT
       26    = IMMINENT DEADLOCK
       41    = DBUPDATE ATTEMPTED TO MODIFY VALUE OF CRITICAL ITEM:
   KEY/SEARCH/SORT
       42    = DBUPDATE WILL NOT ALTER A READ-ONLY DATA ITEM
       43    = DUPLICATE KEY VALUE N MASTER
       44    = CAN'T DELETE A MASTER ENTRY WITH NON-EMPTY DETAIL CHAINS
       49    = ILLEGAL BUFFER ADDRESS
       50    = USER'S BUFFER IS TOO SMALL FOR REQUESTED DATA
       60    = DATABASE ACCESS DISABLED
       61    = PROCESS HAS THE DATABASE OPEN 63 TIMES; NO MORE ALLOWED
       69    = BAD DATABASE
       401:: DBOPEN
       0     = SUCCESSFUL EXECUTION - NO ERROR
       -11   = BAD DATABASE REFERENCE.
       -13   = MUST BE CREATOR OF ROOT FILE OR DATABASE.
       -21   = BAD PASSWORD.
       -22   = MAINTENANCE WORD REQUIRED.
       -31   = BAD MODE.
       -32   = UNOBTAINABLE MODE.
       -34   = DATABASE MUST BE RECOVERED BEFORE ACCESS IS ALLOWED.
       -90   = ROOTFILE BAD.
       -94   = DATABASE BAD.
       60    = DATABASE ACCESS DISABLED.
       61    = THIS DATABASE OPENED MORE THAN 63 TIMES BY THE SAME PROCESS.
       402:: DBINFO
       0     = SUCCESSFUL EXECUTION - NO ERROR
       -11   = BAD DATABASE REFERENCE.
       -21   = BAD DATA SET REFERENCE.
       -31   = BAD MODE.
       -222  = ONLY DBXUNDO ALLOWED WHEN A DYNAMIC TRANSACTION ENCOUNTERS AN
   ERROR.
       49    = ILLEGAL BUFFER ADDRESS.
       50    = BUFFER IS TOO SMALL.
       403:: DBCLOSE
       0     = SUCCESSFUL EXECUTION - NO ERROR
       -11   = BAD DATABASE REFERENCE.
       -21   = BAD DATA SET REFERENCE.
       -31   = BAD MODE.
       -222  = ONLY DBXUNDO ALLOWED WHEN A DYNAMIC TRANSACTION ENCOUNTERS AN
   ERROR.
```

```
   -232  = ILLEGAL DBCLOSE MODE 2 USED DURING AN ACTIVE DYNAMIC TRANSACTION.
   -235  = DYNAMIC TRANSACTION ABORTED DUE TO DBCLOSE MODE 1; DATABASE
CLOSED.
   -420  = FEATURE NO IMPLEMENTED.
   404:: DBFIND
   0     = SUCCESSFUL EXECUTION - NO ERROR
   -11   = BAD DATABASE REFERENCE.
   -21   = BAD DATA SET REFERENCE.
   -31   = BAD MODE.
   -51   = BAD LIST LENGTH.
   -52   = BAD ITEM.
   -222  = ONLY DBXUNDO ALLOWED WHEN A DYNAMIC TRANSACTION ENCOUNTERS AN
ERROR.
   -258  = INVALID ARGUMENT FOR INDEX.
   -259  = INVALID MODE FOR INDEX.
   -260  = NO PREVIOUS LIST OF QUALIFIED DATA ENTRIES.
   17    = NO MASTER ENTRY.
   405:: DBGET
   0     = SUCCESSFUL EXECUTION - NO ERROR
   -11   = BAD DATABASE REFERENCE.
   -21   = BAD DATA SET REFERENCE.
   -31   = BAD MODE.
   -51   = BAD LIST LENGTH.
   -52   = BAD LIST OR BAD ITEM.
   -222  = ONLY DBXUNDO ALLOWED WHEN A DYNAMIC TRANSACTION ENCOUNTERS AN
ERROR.
   10    = BEGINNING OF FILE.
   11    = END OF FILE.
   12    = DIRECTED BEGINNING OF FILE.
   13    = DIRECTED END OF FILE.
   14    = BEGINNING OF CHAIN/QUALIFIER ENTRIES.
   15    = END OF CHAIN/QUALIFIER ENTRIES.
   17    = NO ENTRY.
   18    = BROKEN CHAIN.
   49    = ILLEGAL BUFFER ADDRESS.
   50    = BUFFER IS TOO SMALL.
   406:: DBUPDATE
   0     = SUCCESSFUL EXECUTION - NO ERROR
   -11   = BAD DATABASE REFERENCE.
   -12   = NO LOCK COVERS THE DATA ENTRY TO BE ADDED.
   -14   = ILLEGAL INTRINSIC IN CURRENT ACCESS MODE.
   -21   = BAD DATA SET REFERENCE.
   -31   = BAD MODE.
   -51   = BAD LIST LENGTH.
   -82   = CIUPDATE IS SET TO DISALLOWED; CANNOT USE CRITICAL ITEM UPDATE.
   -222  = ONLY DBXUNDO ALLOWED WHEN A DYNAMIC TRANSACTION ENCOUNTERS AN
ERROR.
   17    = NO ENTRY.
   41    = DBUPDATE ATTEMPTED TO MODIFY VALUE OF CRITICAL ITEM:
KEY/SEARCH/SORT.
   42    = READ ONLY ITEM.
   49    = ILLEGAL BUFFER ADDRESS.
   50    = BUFFER TOO SMALL.
   407:: DBPUT
   0     = SUCCESSFUL EXECUTION - NO ERROR
   -11   = BAD DATABASE REFERENCE.
   -12   = NO LOCK COVERS THE DATA ENTRY TO BE ADDED.
   -14   = ILLEGAL INTRINSIC IN CURRENT ACCESS MODE.
   -21   = BAD DATA SET REFERENCE.
   -23   = DATA SET NOT WRITABLE.
   -24   = OPERATION NOT ALLOWED ON AUTOMATIC MASTER DATA SET.
   -31   = BAD MODE.
   -51   = BAD LIST LENGTH.
```

```
    -52   = BAD LIST OR BAD ITEM.
    -53   = MISSING SEARCH OR SORT ITEM.
    -222  = ONLY DBXUNDO ALLOWED WHEN A DYNAMIC TRANSACTION ENCOUNTERS AN
ERROR.
    16    = DATA SET FULL.
    18    = BROKEN CHAIN.
    43    = DUPLICATE KEY ITEM VALUE.
    408:: DBDELETE
    0     = SUCCESSFUL EXECUTION - NO ERROR
    -11   = BAD DATABASE REFERENCE.
    -12   = NO LOCK COVERS THE DATA ENTRY TO BE DELETED.
    -14   = ILLEGAL INTRINSIC IN CURRENT ACCESS MODE.
    -21   = BAD DATA SET REFERENCE.
    -24   = DBDELETE NOT ALLOWED ON AUTO MASTER.
    -31   = BAD MODE.
    -222  = ONLY DBXUNDO ALLOWED WHEN A DYNAMIC TRANSACTION ENCOUNTERS AN
ERROR.
    17    = NO ENTRY.
    44    = CANNOT DELETE MASTER ENTRY WITH NON-EMPTY DETAIL CHAINS.
    409:: DBLOCK
    0     = SUCCESSFUL EXECUTION - NO ERROR
    -11   = BAD DATABASE REFERENCE.
    -31   = BAD MODE VALUE.
    -121  = DESCRIPTOR COUNT ERROR.
    -123  = ILLEGAL RELOP IN A DESCRIPTOR.
    -124  = DESCRIPTOR TOO SHORT. MUST BE GREATER THAN OR EQUAL TO 9.
    -125  = BAD SET NAME/NUMBER.
    -126  = BAD ITEM NAME/NUMBER.
    -127  = ATTEMPT TO LOCK USING A COMPOUND ITEM.
    -128  = VALUE FIELD TOO SHORT IN A DESCRIPTOR.
    -129  = P-TYPE ITEM LONGER THAN P28 SPECIFIED.
    -130  = ILLEGAL DIGIT IN A P-TYPE VALUE.
    -131  = LOWERCASE CHARACTERS IN TYPE U VALUE.
    -132  = ILLEGAL DIGIT IN TYPE Z VALUE.
    -133  = ILLEGAL SIGN IN TYPE Z VALUE.
    -134  = TWO DESCRIPTORS CONFLICT.
    -135  = DBLOCK CALLED WHEN LOCKS ALREADY IN EFFECT.
    -136  = DESCRIPTOR LIST EXCEEDS 4094 BYTES.
    -222  = ONLY DBXUNDO ALLOWED WHEN A DYNAMIC TRANSACTION ENCOUNTERS AN
ERROR.
    20    = DATABASE LOCKED OR CONTAINS LOCKS.
    22    = DATA SET LOCKED BY ANOTHER PROCESS.
    23    = ENTRIES LOCKED WITHIN SET.
    24    = ITEM CONFLICTS WITH CURRENT LOCKS.
    25    = ENTRY OR ENTRIES ALREADY LOCKED.
    26    = LOCK NOT PERFORMED SINCE DEADLOCK WOULD OCCUR.
    410:: DBUNLOCK
    0     = SUCCESSFUL EXECUTION - NO ERROR
    -11   = BAD DATABASE REFERENCE.
    -31   = BAD MODE.
    -222  = ONLY DBXUNDO ALLOWED WHEN A DYNAMIC TRANSACTION ENCOUNTERS AN
ERROR.
    411:: DBCONTROL
    0     = SUCCESSFUL EXECUTION - NO ERROR
    -11   = BAD DATABASE REFERENCE.
    -14   = ILLEGAL INTRINSIC IN CURRENT ACCESS MODE.
    -31   = BAD MODE.
    -222  = ONLY DBXUNDO ALLOWED WHEN A DYNAMIC TRANSACTION ENCOUNTERS AN
ERROR.
    -224  = DBCONTROL MODE 1 NOT ALLOWED INSIDE A DYNAMIC TRANSACTION.
    412:: DBBEGIN
    413:: DBEND
    414:: DBMEMO
    418:: DBEXPLAIN
    419:: DBERROR
    420:: DBXBEGIN
    0     = SUCCESSFUL EXECUTION - NO ERROR
    -11   = BAD DATABASE REFERENCE.
    -31   = BAD (UNRECOGNIZED) DBXBEGIN MODE.
    -139  = INVALID NUMBER OF BASE IDS.
    -140  = BAD BASE ID LIST.
    -151  = TEXT LENGTH GREATER THAN 512 BYTES.
    -152  = DBXBEGIN CALLED WHILE A TRANSACTION IS IN PROGRESS.
    -217  = DBOPEN MODE INCOMPATIBLE WITH DYNAMIC ROLLBACK.
    -221  = CANNOT BEGIN TRANSACTION WHEN A DYNAMIC TRANSACTION IS ACTIVE.
    -222  = ONLY DBXUNDO ALLOWED WHEN A DYNAMIC TRANSACTION ENCOUNTERS AN
ERROR.
    421:: DBXEND
    0     = SUCCESSFUL EXECUTION - NO ERROR
    -11   = BAD DATABASE REFERENCE.
    -31   = BAD (UNRECOGNIZED) DBXBEGIN MODE.
    -140  = BAD BASE ID LIST.
    -151  = TEXT LENGTH GREATER THAN 512 BYTES.
    -217  = DBOPEN MODE INCOMPATIBLE WITH DYNAMIC ROLLBACK.
    -222  = ONLY DBXUNDO ALLOWED WHEN A DYNAMIC TRANSACTION ENCOUNTERS AN
ERROR.
    -223  = CANNOT DBXEND OR DBXUNDO A TRANSACTION WHICH WAS NOT ACTIVE.
    -238  = MDBX DBXBEGIN, DBXEND MODE MISMATCH.
    422:: DBXUNDO
    0     = SUCCESSFUL EXECUTION - NO ERROR
    -11   = BAD DATABASE REFERENCE.
    -31   = BAD (UNRECOGNIZED) DBXBEGIN MODE.
    -140  = BAD BASE ID LIST.
    -151  = TEXT LENGTH GREATER THAN 512 BYTES.
    -223  = CANNOT DBXEND OR DBXUNDO A TRANSACTION WHICH WAS NOT ACTIVE.
    -238  = MDBX DBXBEGIN, DBXEND MODE MISMATCH.
    -240  = MDBX MODE MISMATCH.
```

Appendix C: TurboIMAGE Data Conversion Reference

TurboIMAGE Data Types

Type	Description
E	ieee floating point. sub-item length is in halfwords
I	signed integer, sub-item length is in halfwords
J	signed integer, but conforms to COBOL standards (i.e. s9999 has max value 9999). sub-item length is in halfwords
K	unsigned integer, no negative value. 1 halfword = 0-65K, 2 halfwords= 0-2 Billion, sub-item length is in halfwords
P	packed decimal, sub-item length is in nibbles, 2 to 28, with one digit used for the sign (note: TurboIMAGE will let you create a P48 or even larger, but COBOL will not process it)
R	classic HP 3000 floating point, old, 2 halfwords or 4 halfwords
U	uppercase ASCII chars, sub-item length is in bytes
X	any ASCII characters, sub-item length is in bytes
Z	zoned decimal number. sub-item length is in bytes

NOTES: The size of the entire data item must be a multiple of halfwords (16 bits). Therefore, P types normally come in multiples of 4 and U/X/Z types come in multiples of 2.

TurboIMAGE Conversion Types

Xn	Character, n bytes, define as Character in FORTRAN, X(n) in COBOL.
Un	Uppercase Character, n bytes, define as Character in Fortran, A(n) in COBOL
E2	Floating-Point, 4 bytes, define as Real in Fortran, not supported in HP COBOL
E4	Floating-point, 8 bytes, define as Double Precision in Fortran, not supported in HP COBOL
I1/J1	Integer, 2 bytes, define as Integer*2 in Fortran, S9 to S9(4) Comp in COBOL
I2/J2	Integer, 4 bytes, define as Integer*4 in Fortran, S9(5) to s9(9) Comp in COBOL
I4/J4	Integer, 8 bytes, define as S9(10) to S9(18) Comp in COBOL, not supported in Fortran
K1	Logical, 2 bytes, define as Logical in Fortran, not supported in COBOL
Zn	Zoned-Decimal, n bytes, s(n) Display in COBOL, overpunched
P4	Packed-Decimal, 2 bytes, s9(3) Comp-3 in COBOL, not supported in Fortran.

P8	Packed-Decimal, 4 bytes, s9(7) Comp-3 in COBOL, not supported in Fortran.
Pn	Packed-Decimal, n/2 bytes, s9(n-1) Comp-3 in COBOL, not supported in Fortran. Maximum N in HP COBOL is 19 (18 digits plus a sign).
Zn	Numeric Display, n bytes, s9(n) Display in COBOL, with sign "overpunched" in the units position (unless you specify SIGN IS SEPARATE, then there are only n-1 digits in the value).

iMaxsoft Specific TurboIMAGE Conversions

DB2	
I4	DB2 Format Load Option cannot handle direct BINARY to INTEGER load, therefore OTDRV converts IMAGE BINARY integer to an ASCII. Since i4 range is from: −9,223,372,036,854,775,808 to +9,223,372,036,854,775,807, it is represented using ASCII characters hence the need for 19-characters (or 19 bytes) to convey the integer information

iMaxsoft

3 easy ways to reach us for support:

- Visit us on the Web
- E-mail us: support@imaxsoft.com
- Call us: 408.253.1987 (24/7, 365)

iMaxsoft Corp.

20410 Town Center Lane, Ste. 295
Cupertino, CA 95014

Phone (408) 253-8808
Fax (408) 253-4008

www.imaxsoft.com