OPENTURBO™ for ORACLE

iMaxsoft

iMaxsoft Corporation

Version B.01.00

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
6/07/2006	1.0	Oliver Wai	Version 1.0

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.

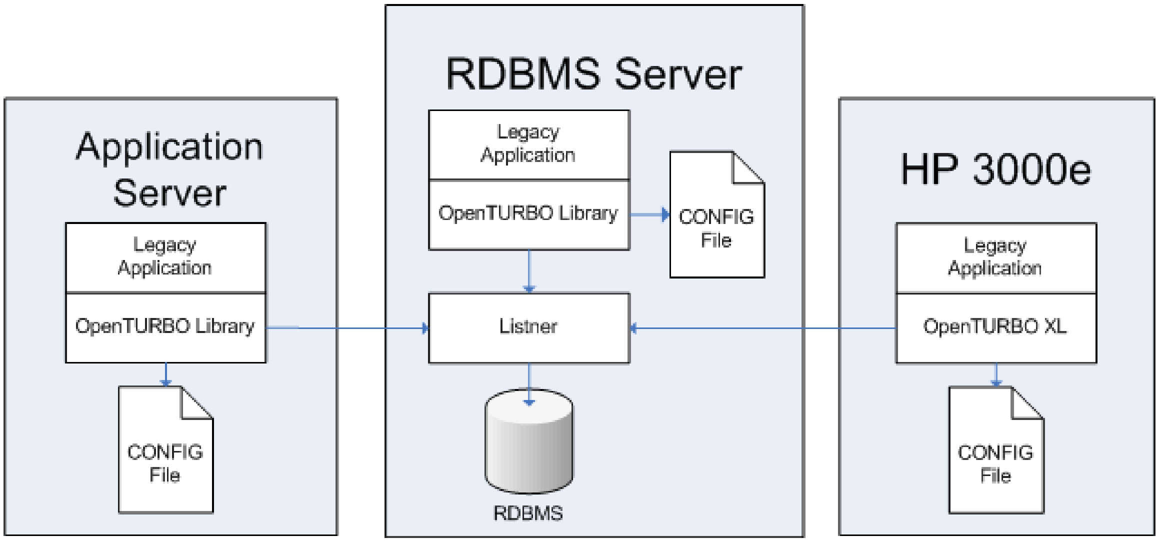

Figure 1: OPENTURBO Configurations

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.

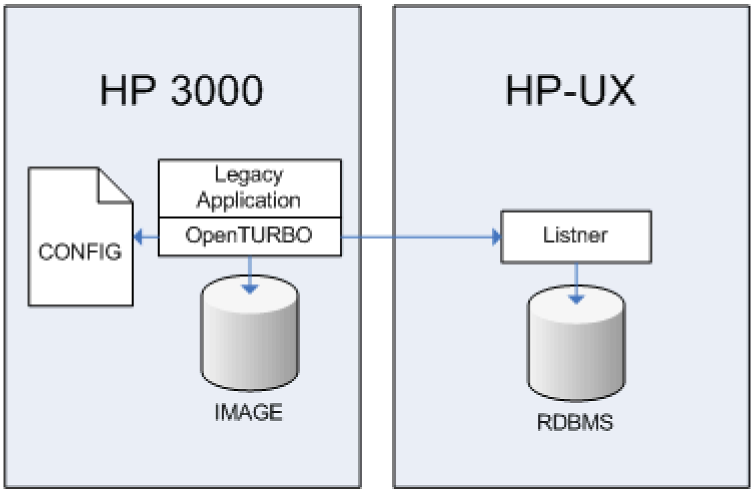

Figure 2: Mirror IMAGE & RDBMS w/legacy application on HP3000

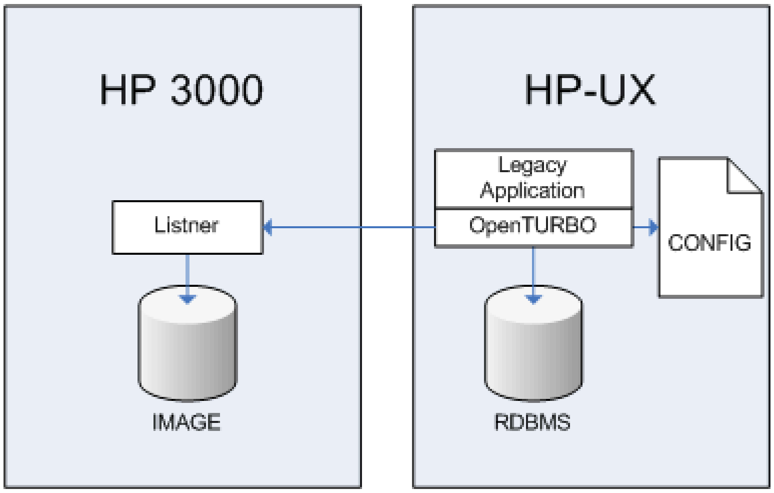

Figure 3: Mirror IMAGE & RDBMS w/ legacy application on HP-UX

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

OPENTURBO for Oracle supports 4 different HP architectures. These include:

- PA-RISC 1.1 (32 bit)
- PA-RISC 1.1 (64 bit)
- Itanium IA64 (32 bit)
- Itanium IA64 (64 bit)

The core OPENTURBO library package consists of the following libraries:

Library	**Description**
libot.sl, libsdkc.sl	Client Server Mode Libraries
libotdbg.sl	Client-Server Mode Debug Library
libotD02.sl libsqlD02.sl (48 cursors) libsqlD02144.sl (144 cursors) libsdkc.sl libsqlcD02.sl	Direct Mode Libraries
libotD02dbg.sl libotD02144dbg.sl	Direct Mode Debug Libraries
libmpe.sl	MPE Intrinsic Library
liblt.sl	Middleware Library
libti.sl, tiam.sl	OPENTURBO root file API Library
libdbutx.sl	Database clean-up library

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 HP 3000e 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
```

CONFIG File Command Syntax

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

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.ORACLE.IMAXSOFT. For Unix servers, the default is: /opt/imaxsoft/OPENTURBOx.x/oracle/conf/RESERVE.ORACLE
OT_ERROR_FILE	Points to the file that contains the TurboIMAGE errors and messages for OPENTURBO. Default for HP3000 OTERROR.ORACLE.IMAXSOFT. For Unix servers, the default is: /opt/imaxsoft/OPENTURBOx.x/oracle/conf/OTERROR.ORACLE
OT_HOST	Points to the application server or the machine where your ORACLE (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_SERVICE is ignored if OPENTURBO is functioning in direct access mode on local database.

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: 515 in decimal form`
	`ELOQUENCE on HP-UX: 521 in decimal form`
	`DB2 on HP-UX: 528 in decimal form`

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: `<ORACLE fully qualified name>/<encrypted password>` If OT_RDB_LOGON is set, then OT_RDB_OWNER must also be set.
OT_RDB_OWNER	The format of this field is: `<ORACLE fully qualified name>` For ORACLE, OT_RDB_OWNER is the database name stored in the ORACLE instance. For example, if the database name is DB1.GROUP.ACCOUNT then OT_RDB_OWNER=DB1. OPENTURBO will

<table>
<tr><td></td><td>replicate DB1.GROUP.ACCOUNT in IMAGE to DBNAME in ORACLE.

If OT_RDB_LOGON is set, then OT_RDB_OWNER must also be set.</td></tr>
<tr><td>OT_SDK_SERVER_PGM</td><td>Contains the program name of OPENTURBO Core Server Program. By default the OPENTURBO uses:

/opt/imaxsoft/OPENTURBOx.x/oracle/bin/dbsvrA02.144

This field only applies to OPENTURBO Client-Server mode.</td></tr>
<tr><td>OT_CIUPDATE</td><td>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).</td></tr>
<tr><td>OT_IMAGEMODE</td><td>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.</td></tr>
<tr><td>OT_DUALMODE</td><td>Different behaviors are set depending on the combination of OT_DUALMODE and OT_IMAGEMODE:

<table>
<tr><th>OS</th><th>IMAGEMODE</th><th>DUALMODE</th><th>Comment</th></tr>
<tr><td>MPE/XL</td><td>ON</td><td>OFF</td><td>TurboIMAGE only</td></tr>
<tr><td>MPE/XL</td><td>-</td><td>ON</td><td>DB2/others RDBMS primary TurboIMAGE secondary</td></tr>
<tr><td>MPE/XL</td><td>OFF</td><td>OFF</td><td>DB2/others RDBMS only</td></tr>
<tr><td>HP-UX</td><td>ON</td><td>OFF</td><td>TurboIMAGE only</td></tr>
<tr><td>HP-UX</td><td>-</td><td>ON</td><td>TurboIMAGE primary DB2/others RDBMS secondary</td></tr>
<tr><td>HP-UX</td><td>OFF</td><td>ON</td><td>DB2/others RDBMS primary TurboIMAGE secondary</td></tr>
<tr><td>HP-UX</td><td>OFF</td><td>OFF</td><td>DB2/others RDBMS only</td></tr>
</table></td></tr>
</table>

	For unidirectional data replication and synchronization across an IMAGE database and Oracle (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`	There are 3 modes of caching used: **`ON`** ORACLE INSERT, UPDATE and DELETE queries are cached. For applications performing duplicate queries multiple times, caching will result in major performance improvements. **`OFF`** No caching. Each INSERT, UPDATE and DELETE queries are processed each time. **`STATEMENT_COMMIT`** When `OT_DUALMODE` is set to `2PC` (2 Phase Commit), data is synchronized across the HP9000 and HP3000 machines. 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.

	If `OT_WRITE_CACHE` is OFF when `OT_DUALMODE=2PC`. OPENTURBO will wait for commit on the remote server after every call before issuing the next call. This results in much slower performance in 2PC mode.
`OT_READ_CACHE`	Similar to `OT_WRITE_CACHE`, ORACLE SELECT queries can be cached for faster performance. **ON** Turns on caching. **OFF** (Default) No caching.
`OT_LOCKWAIT_CYCLE`	Controls the number of wait cycles when lock conflict occurs and unconditional LOCK option is used. The default number of cycles is `OT_LOCKWAIT_CYCLE = 0`, which waits forever. Each wait cycle interval is 2 seconds. For example, if you set the `OT_LOCKWAIT_CYCLE = 60`, the OPENTURBO Lock Mmanager will try to grant lock within 2 minutes (60 x 2 = 120 seconds) before giving up and return back to your application.
`OT_NETWORK_COMPRESS`	Used to speed up network transmission and is only applicable to OPENTURBO Client-Server mode. OPENTURBO will trim trailing blanks before send and pad back the trailing blanks before returning to your program. **ON** For getting data from ORACLE servers to your applications only. It trims TRAILING BLANKS only for all X, U and Z IMAGE datatypes equivalent columns. It has no effects when sending data to the ORACLE server. **OFF** (Default) No trimming of whitespace.
`OT_LOCKCOVERAGE`	Used to override the pre-update re-fetch stand feature. It is not recommended for OLTP and multi-users apps. It is intended for single user batch process where there is exclusive database access. **ON** (Default). OPENTURBO functions normally. **OFF** OPENTURBO will not perform LOCK COVERAGE checking for DBUPDATE and DBDELETE. The default pre-fetch and no before-

	update image comparisons are ignored.
OT_DBLOCK_CONTROL	OPENTURBO supports 2 additional options other than the standard default (OPTION 0). **OPTION = 0** (Default) Uses the default IMAGE locking mechanism **OPTION = 1** Item Level (Predicate Level or Row Level) locking strategy is used; this method uses DBLOCK2 table only, and a UNIQUE CONSTRAINT of (DBASE, DSET, ITEN, VAL) must be created for table DBLOCK2. This option is application specific; you must be 100% sure that only predicate lock is used through-out your entire applications. **OPTION = 9** No locks are used. This option is intended in the case of batch programs where the batch job has exclusive access to the database (i.e. overnight jobs). In this situation, there is no need to lock/unlock the resource. Turning off locking mechanism results in major performance boosts.
OT_RDB_SID	Reserved for future use

Dataset Level Configuration:

Dataset level configurations are options that iMaxsoft has developed for OPENTURBO on ORACLE to improve overall performance. Depending on the nature of your application and/or dataset, modifying some of the dataset level configurations can result in major performance and throughput improvements. These fields include:

- OT_BULKCHAINGET
- OT_IGNORE_CHAINSTATUS
- OT_IGNORE_DBPUTSTATUS
- OT_SERIAL_ORDERBY
- OT_TRIM_WHITESPACE

Field	**Description**
OT_DETAILSETNAME	Specifies the name of a detail dataset. *NOTE: '@' is for all detail datasets.*
OT_BULKCHAINGET	*Option is ignored if* *OT_IGNORE_CHAINSTATUS = ON* OT_BULKCHAINGET option supports one direction chain get only,

	either DBGET Mode 5 forward or DBGET Mode 6 backward, but not both. The reason for only unidirectional support is that unidirectional support has a lower overhead resulting in major performance improvements. **ON** Triggers OPENTURBO bulk fetch and cache features. OPENTURBO gets as many rows as it can and put them into the internal cache buffer **OFF** (Default Behavior) OPENTURBO gets only one row at a time.
OT_IGNORE_CHAINSTATUS	**ON** OT_BULKCHAINGET is ignored by OPENTURBO. OPENTURBO will perform either forward (MODE 5) or backward (MODE 6) chain get without tracking chain-count, forward and backward pointers. Dataset access does not have to reply on any forward and backward chain pointers result in major performance improvements over the standard forward/backward chains. If one of your applications does only one direction chain get and does not use status words 5-10, then you should use this option to significantly improve through-put and performance **OFF** (Default behavior). Use OT_BULKCHAINGET
OT_CHRONOLOGICAL	OT_CHRONOLOGICAL is applicable for DBPUT, DBUPDATE and DBDELETE. **ON** OPENTURBO sorts IMAXSOFT13_PATH_nn and IMAXSOFT13_SEQ_NO in ascending order for DBGET Mode 5 and in descending order for DBGET Mode 6. In DBPUT, this option will force OPENTURBO to maintain the chronological order of each IMAXSOFT13_PATH_nn. **OFF** (Default behavior) IMAGE chronology will not be kept in ORACLE
OT_IGNORE_DBPUTSTATUS	Most applications ignore the returned record number. Performance can be improved by suppressing the return of record number. **ON** OPENTURBO returns no record number after successful DBPUT. This is strictly for performance.

	OFF (Default behavior) OPENTURBO returns record number.
OT_USE_IMAGERECNUM	*Option applies on when OT_DUAL_MODE=ON* **ON** OPENTURBO will DBPUT to TurboIMAGE first, then use the returned record number for ORACLE INSERT statement. This is to insure an exact mirror of TurboIMAGE and ORACLE (in sequence) **OFF** IMAGE and ORACLE record sequence will not be in sync
OT_SERIALGET_ORDERBY	The default DBGET Mode 2 reads rows in series (i.e. 2 or 3 in IMAGE) from a table based on the database engine internal orders. This is used to ensure consistent sequential table access in ORACLE. Since most applications don’t care about the order of the results, performance can be greatly improved by turning this option OFF. **ON** Guarantees the sequential order on ORACLE by using IMAXSOFT13_SEQ_NO **OFF** Returns records in native order
OT_TRIM_WHITESPACE	Trims whitespace at the end of fields. This calls impacts ALL TurboIMAGE calls. Improper settings will cause major database integrity issues. **ON** In order to use OT_TRIM_WHITESPACE=ON, you must: 1. You must generate ORACLE schema with option –z (OTDRV "... -t2 -z"), which allows for NULL fields in VARCHAR2 columns. 2. You must generate ORACLE sqlldr CLT script with option –z (OTDRV "... -t20 -z"), which generates load scripts that enable ORACLE sqlldr to load all BLANKS as NULL and to trim trailing BLANKS **OFF** (Default) No trimming of whitespaces.

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]
```

OPENTURBO Synchronization

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. In addition if the synchronization direction is from from HP-UX to the HPe3000 where the TurboIMAGE resides, then you must set `TI_DUALMODE_HOST, TI_DUALMODE_SERVICE, TI_DUALMODE_PGM` in the configuration file.

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/oracle/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/oracle/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/oracle/config/mynice

[ Q1 ]
   SERVER = /opt/imaxsoft/OPENTURBO3.7/oracle/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/oracle/myprog1 30 /opt/imaxsoft/OPENTURBO3.7/oracle/myprog2 19 /opt/imaxsoft/OPENTURBO3.7/oracle/myprog3 20 /opt/imaxsoft/OPENTURBO3.7/oracle/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.[1]

3. Set values for CONFIG files.

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

[1] Not applicable if running OPENTURBO only on local database.

```
DBNAME.GROUP.ACCOUNT {
DBNAME.GROUP {
DBNAME {
  OT_IMAGEMODE         = OFF
  OT_ROOT_FILE         = DBNAMEti.ti
  OT_RESERVE_WORD_FILE = RESERVE.ORACLE
  OT_ERROR_FILE        = OTERROR.ORACLE
  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/oracle/bin/dmdrv
  OT_CIUPDATE          = ON
  OT_DUALMODE          = 2PC
  OT_TRX_THRESHOLD     = 2
  OT_WRITE_CACHE       = ON
  OT_READ_CACHE        = ON

OT_DETAILSETNAME = @ [
   OT_IGNORE_CHAINSTATUS  = OFF
   OT_BULKCHAINGET        = ON
   OT_IGNORE_DBPUTSTATUS  = ON
   OT_USE_IMAGERECNUM     = ON
   OT_CHRONOLOGICAL       = ON
   OT_SERIALGET_ORDERBY   = ON
   OT_TRIM_WHITESPACE     = OFF
]
}
```

From HP-UX to HPe3000

1. Make sure ORACLE 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[2]

4. Set values for the CONFIG files.

```
OT_TI_DBNAME = DBNAME.GROUP.ACCT
DBNAME.GROUP.ACCT {
DBNAME.GROUP {
DBNAME {
   OT_ROOT_FILE         =
/opt/imaxsoft/OPENTURBO3.7/oracle/db/inventti.ti
   OT_RESERVE_WORD_FILE =
/opt/imaxsoft/OPENTURBO3.7/oracle/conf/RESERVE.ORACLE
   OT_ERROR_FILE        =
/opt/imaxsoft/OPENTURBO3.7/oracle/conf/OTERROR.ORACLE
   OT_HOST              = 207.92.64.8
   OT_SERVICE           = 32601
```

[2] Not applicable if running OPENTURBO only on local database.

```
   OT_OS_RDBMS          = 515
   OT_RDB_LOGON         = DBNAME_GROUP_ACCOUNT/YWIVHZ
   OT_RDB_OWNER         = AMISYS
   OT_SDK_SERVER_PGM    =
/opt/imaxsoft/OPENTURBO3.7/oracle/lbin/dbsvrA02
   OT_CIUPDATE          = ON
   OT_DUALMODE          = 2PC
   TI_DUALMODE_HOST     = 207.92.64.66
   TI_DUALMODE_SERVICE  = 32602
   TI_DUALMODE_PGM      = DMDRV.BIN.IMAXSOFT
   OT_LOCKWAIT_CYCLE    = 60
   OT_TRX_THRESHOLD     = 1
   OT_NETWORK_COMPRESS  = ON
   OT_PRERELEASE        = A0201
// For INSERT Caching
   OT_DBLOCK_CONTROL    = 0
// 1 - ITEM Level and = Operator only, Unique constraint for DBLOCK2
//     (DBASE, DSET, ITEM, VAL)
   OT_LOCKCOVERAGE      = ON
   OT_WRITE_CACHE       = OFF
   OT_READ_CACHE        = OFF
   OT_NO_SETCOUNT       = OFF
OT_DETAILSETNAME = @ [
   OT_IGNORE_CHAINSTATUS  = OFF
   OT_BULKCHAINGET        = ON
   OT_IGNORE_DBPUTSTATUS  = ON
   OT_USE_IMAGERECNUM     = ON
   OT_CHRONOLOGICAL       = ON
   OT_SERIALGET_ORDERBY   = ON
   OT_TRIM_WHITESPACE     = OFF
]
//
// For Detail Dataset MYSET Only
//
OT_DETAILSETNAME = MYSET [
   OT_IGNORE_CHAINSTATUS  = OFF
   OT_BULKCHAINGET        = ON
   OT_IGNORE_DBPUTSTATUS  = ON
   OT_USE_IMAGERECNUM     = ON
   OT_CHRONOLOGICAL       = ON
   OT_SERIALGET_ORDERBY   = ON
   OT_TRIM_WHITESPACE     = OFF
]
}
```

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>
```

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 both the ORACLE and OPENTURBO libraries. In addition the OPENTURBO library path ***MUST*** come before the ORACLE path. For example:

```
export SHLIB_PATH=/opt/imaxsoft/OPENTURBO3.7/lib:$ORACLE_HOME/lib32
```

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 2006`
Check product number 2006 and 1688

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.

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**	**ChangeTo**
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	ChangeTo	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	AMIOWN	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`, if you need to use CROSSREF 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.oracle -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.oracle &
-s si &
-recnum -b ORACLE -e crossref.config &
-g 207.92.64.9:32688:INVENT:INVENT:/opt/imaxsoft/oracle/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.oracle
          -t20
          -rinvent18.OTDATA.IMS0100
          -linvent18.OTSCRIPT.IMS0100
          -oinvent18.OTCOPY.IMS0100
          -winvent18.OTWORK.IMS0100
          -sSI
          -bORACLE
          -ecrossref.config
           EXCEPTION REPORT = invent18.OTEXCEPT.IMS0100
           OTEDIT_OFF
           CHRONOLOGICAL_OFF
           ASCII_TRANSFER
           NON_EXCLUSIVE_MODE
           USE_TI_RECNUM and USE_FOR_DOOR
           ORACLE
           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/oracle/otldr2
              LOGFILE dir     =/tmp/
              COMMIT Count    =2000

ORACLE:  TOTLen         = 4076
ORACLE:  RDBFileRecSize = 4076
ORACLE:  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 ORACLE 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.

listners

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/oracle/listner 32601
```

You need to start the listener with super user capability and from the login with proper ORACLE and OPENTURBO setup, which means all environment variables, access paths, library paths, etc. must point to the proper places for ORACLE database and OPENTURBO, MF-COBOL, ORACLE 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/pub:/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/oracle/myprog1 30`
	`/opt/imaxsoft/OPENTURBO3.7/oracle/myprog2 19`

	/opt/imaxsoft/OPENTURBO3.7/oracle/myprog3 20 /opt/imaxsoft/OPENTURBO3.7/oracle/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/oracle/config/mynice

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

otANALYZER

Produces a series of reports to validate the integrity of your ORACLE database, to analyze your ORACLE database for performance enhancements, and to check the accuracy of OPENTURBO ORACLE migration.

NOTE: All passwords must be entered in their encrypted form.

```
OPENTURBO otANALYZER <A.01.00> iMAXSOFT Corp. Copyright 2002

usage:otALANYZER -u -d -r -o
      -u ORACLE Login User/Password
      -d TurboIMAGE Fully Qualified Database Name
         DBName.GROUP.ACCOUNT
      -r Report ID, you may specify one or many separated
         by comma; such as -r1 or -r1,2,3
         Report ID Report Description
         ========= ========================================
             1     Detail Dataset Chain Length Analysis
             2     ORACLE Tablespaces Allocation Analysis
             3     Post-Migration Dataset Count Validation
      -o Report Output Filename, Default = Report
```

Example

```
/opt/imaxsoft/OPENTURBO3.7/oracle/bin/otANALYZER -d MUSIC.TIDATA.IMAXSOFT \
                          -u ot/JO \
                          -r 1,2,3 \
                          -o
/opt/imaxsoft/OPENTURBO3.7/oracle/musicdemo/report/DBReport
```

This will produce the following output

```
Report 1 – Detail Dataset Path Analysis:

 Report  : PATH ANALYSIS
 ImageDB : MUSIC.TIDATA.IMAXSOFT                            Date: July 14 2002
 Owner   : MUSIC_TIDATA_IMAXSOFT

                                                     Path Length Statistic
     Table                   Index Column            Min   Max/BOF  Avg/Total
 ----------------------- ----------------------- -------- -------- ----------
 ACCOUNT_FILE            ACCOUNT                        1        1       1.00
                                                 ----------------- ----------
                                                                 1        955*
                                                 ----------------- ----------
 ACCOUNT_FILE2           ACCOUNT                        1        1       1.00
                                                 ----------------- ----------
                                                                 1        415*
                                                 ----------------- ----------
 ACCOUNT_FILE3           ACCOUNT                        1        1       1.00
                                                 ----------------- ----------
                                                                 1        239*
                                                 ----------------- ----------
 SEARCH_KEY_FILE         SEARCH_KEY                     1        1       1.00
                                                 ----------------- ----------
1       761*
```

```
// … (continues onwards)

       Report 2 – ORACLE Tablespace Requirement Analysis:

 Report  : TABLESPACE REQUIREMENT
 ImageDB : MUSIC.TIDATA.IMAXSOFT                         Date: July 14 2002
 Owner   : MUSIC_TIDATA_IMAXSOFT

     Table               Tablespace(DATA) kbytes  Tablespace(INDEX) kbytes
 ----------------------- ------------------------ -------------------------
 ACCOUNT_FILE                                  12                        10
 ACCOUNT_FILE2                                  5                         4
 ACCOUNT_FILE3                                  3                         2
 SEARCH_KEY_FILE                               15                         8
 SSN_FILE                                      14                        10
 AUTO_KEY_X8                                    0                         0
 MEMBER_FILE                                  960                        30
 ACC_FILE                                       4                         1
 NEXT_ACC_FILE                                  0                         0
 SHARE_FILE                                  1502                        71
 IRA_FILE                                      94                         9
 LOAN_FILE                                   1300                        31
 COLLATERAL_FILE                               17                         2
 LN_PUR_FILE                                    2                         1
// … (continues onwards)

       Report 3 – OPENTURBO Load Analysis:

 Report  : SEQUENCE NO
 ImageDB : MUSIC.TIDATA.IMAXSOFT                         Date: July 14 2002
 Owner   : MUSIC_TIDATA_IMAXSOFT

     Table               TYPE   IMAXSOFT13_SEQ_NO      SEQUENCE INITIAL #
 ----------------------- ---- ---------------------- ----------------------
 ACCOUNT_FILE               A                    955                    955
 ACCOUNT_FILE2              A                    415                    415
 ACCOUNT_FILE3              A                    239                    239
 SEARCH_KEY_FILE            A                    761                    761
 SSN_FILE                   A                    971                    971
 AUTO_KEY_X8                A                      6                      6
 MEMBER_FILE                D                    913                    913
 ACC_FILE                   M                     35                     35
 NEXT_ACC_FILE              M                      0                      0
 SHARE_FILE                 D                   1900                   1900
 IRA_FILE                   D                    282                    282
 LOAN_FILE                  D                   1103                   1103
 COLLATERAL_FILE            M                     75                     75
// … (continues onwards)
```

otCLEAN

Deletes Automatic table entries that are not foreign to any Detail tables. OPENTURBO delete function doesn’t delete keys without details automatically, so you must run this program periodically to clean-up any dangling keys from Automatic tables. The reason is performance.

TurboIMAGE Automatic dataset has absolutely no value in the ORACLE environment, OPENTURBO maps them into ORALCE indexes which serve the sole purpose of fast key search; unless you have application requires serial scan of those Automatic datasets, otherwise, do not migrate Automatic datasets.

```
OPENTURBO otCLEAN <A.01.03> iMaxsoft Corp. Copyright 2002

usage: otCLEAN -u -d -r -o
       -u ORACLE OPENTURBO Superuser Login OT/Password
       -d TurboIMAGE Fully Qualified Database Name
          DBName.GROUP.ACCOUNT - to be cleaned
       -r 1 = Clean Up Automatic Tables Only
          2 = Reserved
          3 = Reserved
       -o Execution Log File Name, Default = otCLEAN.log
```

NOTE: All passwords must be entered in their encrypted form.

Example:

```
: otCLEAN -d MUSIC.TIDATA.IMAXSOFT \
                        -u ot/JO \
                        -r 1 \
                        -o
/opt/imaxsoft/OPENTURBO3.7/oracle/musicdemo/log/MUSICClean.log
```

otCHECKChecksum

Compares checksums in ORACLE database and the OPENTURBO TIFile to ensure data integrity between the IMAGE database and the migrated ORACLE database.

```
OPENTURBO otCHECKChecksum <A.01.03> iMaxsoft Corp. Copyright 2002

usage: otCHECKChecksum -u -d -t
       -u ORACLE Login User/Password
       -d TurboIMAGE Fully Qualified Database Name
          DBName.GROUP.ACCOUNT.
       -t TIFile Name
```

Example

```
/opt/imaxsoft/OPENTURBO3.7/oracle/bin/otCHECKChecksum -u OT/JO \
                               -d MUSIC.TIDATA.IMAXSOFT \
                               -t ../tidb/ti

OPENTURBO otCHECKChecksum <A.01.03> iMaxsoft Corp. Copyright 2002

oracle_user  - [OT/JO]
tibase_name  - [MUSIC.TIDATA.IMAXSOFT]
tifile       - [../tidb/ti]

Checksum Summary Report:

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

otCHECKChecksum Result = 1
                     1 = Matched
                     0 = Not Matched
                    -1 = Error
```

otDBCOPY

Makes a copy of a logical TurboIMAGE database in ORACLE environment. If -x is used, three script files will be generated, `OTCopyDB_test.EXECUTE`, `OTCopyDB_test.SQL` and `OTCopyDB_test.CLASS`. You must manually start `OTCopyDB_text.EXECUTE` to copy a database. If -x is omitted, the database will be copied in real-time.

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

usage: otDBCOPY -u -d -t -r -n -o -p -s -x
       -u ORACLE Login User/Password
       -d TurboIMAGE Fully Qualified Database Name
          DBName.GROUP.ACCOUNT
       -t Old TIFile Name
       -r ORACLE RESERVE WORD File Name
       -n New Database Name
       -o New TIFile Name
       -p New ORACLE User DBName_GROUP_ACCOUNT Password
       -s Default Tablespace for storing New ORACLE User
          Security Objects that are replicated from
          TurboIMAGE Access Class and Password
       -x Output ORACLE Script File Name
```

Notes

- All passwords must be entered in their encrypted form.
- Options –d and –t must contain database names that are MPE compliant, which is max of 6 characters for database name, max of 8 characters for group and max of 8 characters for account, all must lead with an alpha character.

Example

```
#export LTDBG1=1
#export LTDBG2=1
#export LTDBGOUT=-
otDBCOPY -u OT/JO            \
         -d MUSIC.DATA.TICHG \
         -t MUSIC.TI \
         -r /opt/imaxsoft/OPENTURBO3.7/oracle/conf/RESERVE.ORACLE \
         -n MUSIC.COPY.TICHG \
         -o NEWTI            \
         -p HPNDX            \
         -s MUSIC_TICHG      \
         -x test
```

otDBPURGE

Purges a logical TurboIMAGE database from an ORACLE instance. You must purge the old database before you can reuse the same database name, a fully qualified TurboIMAGE database name. If -x is used, two script files will be generated, `OTPurgeDB_test.EXECUTE` and `OTPurgeDB_test.SQL`, you must manually start `OTPurgeDB_text.EXECUTE` to purge the database. If -x is omitted, the database will be purged in real-time.

```
OPENTURBO otDBPURGE <A.01.03> iMaxsoft Corp. Copyright 2002

usage: otDBPURGE -u -t -p -d -x
       -u ORACLE Login User/Password
       -d TurboIMAGE Fully Qualified Database Name
          DBName.GROUP.ACCOUNT
       -t TIFile Name
       -p New ORACLE User DBName_GROUP_ACCOUNT Password
       -x Output ORACLE Script File Name
```

Note: All passwords must be entered in their encrypted form.

Example

```
otDBPURGE -u OT/JO            \
          -d MUSIC.COPY.TICHG \
          -t /opt/imaxsoft/OPENTURBO3.7/oracle/otb/tdrv/MUSIC.COPY.TICHG \
          -p HPNDX            \
          -x test
```

otDBUTIL

DBUTIL is subset of TurboIMAGE DBUTIL tool. It lists DBOPEN processes and DBLOCK requests from your ORACLE database to the same TurboIMAGE logical database. This utility can be run on the net and uses the same CONFIG file as other OPENTURBO libraries and utilities. The CONFIG can be re-directed via file equation or environment variable OT_CONFIG.

Example:

```
       :FILE CONFIG=CONFIG.LEE.IMAXSOFT
       :otDBUTIL

OPENTURBO DBUTIL <A.01.02> iMaxsoft Corp. Copyright 2002.

>>help

EXIT                      : Leave the program
SHOW (database-name) USERS: Display all current OPENTURBO users
SHOW (database-name) LOCKS: Display all current OPENTURBO locks

>>show invent users

For database INVENT.DATA.MOULTON

PIN   PATH      EXECUTING PROGRAM      JOBNUM MODE
===== ==== =========================== ====== ====
 2227  1           OPENTURBO DBOPEN            1
 2254  1           OPENTURBO DBOPEN            1

>>show invent locks

For database INVENT.DATA.MOULTON

LOCKED ENTITY                                     PIN       PROGRAM
================================================ ===== =================
DATA SET    ITMMAST . . . . . . . . . . .         2227 OPENTURBO DBLOCK
    ITMDTL:  CC-PARTNO  =   MOXIMG14047          2254  OPENTURBO DBLOCK

>>
```

otgenDB

Converts the ORACLE schema file created from HP3000 into ORACLE sqlplus format and adds security.

```
   OPENTURBO otgenDB <A.01.03> iMaxsoft Corp. Copyright 2002

   usage: otgenDB -d -u -k -p -s -f -x
          -u ORACLE Login User/Password
          -d TurboIMAGE Fully Qualified Database Name
             DBName.GROUP.ACCOUNT
          -k Input ORACLE Schema File from HP/3000
          -p ORACLE User DBName_GROUP_ACCOUNT Password
          -s Default Tablespace for storing New ORACLE User
             Security Objects that are replicated from
             TurboIMAGE Access Class and Password
          -x Output ORACLE Script File Name
          -f Use this option when this is the very first time
             to create the specified TurboIMAGE database into
             target ORACLE instance, which means the TurboIMAGE
             database never exist in the target ORACLE.
```

Note: All passwords must be entered in their encrypted form.

Example

```
otgenDB -uOT/JO \
        -dMUSIC.TIDATA.IMAXSOFT \
        -k../data/SMUSIC2.OTSCHEMA \
        -pHPNDX \
        -sMUSIC
```

The above example will product an output file `OTBuildDB.EXECUTE`. This file will:

1. Drop all tables and MUSIC.TIDATA.IMS related objects from ORACLE database (OTBuildDB.DROP)

```
ALTER TABLE MUSIC_TIDATA_IMAXSOFT.ALBUMS         DROP PRIMARY KEY CASCADE;
DROP  TABLE MUSIC_TIDATA_IMAXSOFT.ALBUMS;
ALTER TABLE MUSIC_TIDATA_IMAXSOFT.COMPOSERS      DROP PRIMARY KEY CASCADE;
DROP  TABLE MUSIC_TIDATA_IMAXSOFT.COMPOSERS;
ALTER TABLE MUSIC_TIDATA_IMAXSOFT.SELECTIONS_A   DROP PRIMARY KEY CASCADE;
DROP  TABLE MUSIC_TIDATA_IMAXSOFT.SELECTIONS_A;
DROP  TABLE MUSIC_TIDATA_IMAXSOFT.SELECTIONS;
DROP  TABLE MUSIC_TIDATA_IMAXSOFT.LOG;
COMMIT WORK;

DELETE FROM OT.TISET        WHERE TIBASE_NAME = 'MUSIC.TIDATA.IMAXSOFT';
DELETE FROM OT.TISET_KEY    WHERE TIBASE_NAME = 'MUSIC.TIDATA.IMAXSOFT';
DELETE FROM OT.TISET_FIELD  WHERE TIBASE_NAME = 'MUSIC.TIDATA.IMAXSOFT';
COMMIT WORK;
EXIT
```

2. Create all MUSIC.TIDATA.IMS tables and objects in ORACLE database (`/opt/imaxsoft/OPENTURBO3.7/oracle/musicdemo/data/SMUSIC2.OTSCHEMA`)

```
CREATE TABLE OT.TISET (
   TIBASE_NAME         VARCHAR2(26) NOT NULL,
   TABLE_NAME         VARCHAR2(256) NOT NULL,
   TISET_NAME          VARCHAR2(16) NOT NULL,
   TISET_NO              NUMBER(10) NOT NULL,
   TISET_TYPE          VARCHAR2(01) NOT NULL,
   LAST_SEQ_NO           NUMBER(20) NOT NULL,
   TIMESTAMP           VARCHAR2(14) NOT NULL
   );

CREATE TABLE OT.TISET_FIELD (
   TIBASE_NAME         VARCHAR2(26) NOT NULL,
   TISET_NO              NUMBER(10) NOT NULL,
   TIFIELD_SEQ_NO        NUMBER(10) NOT NULL,
   COLUMN_NAME        VARCHAR2(256) NOT NULL,
   TIITEM_NAME         VARCHAR2(20) NOT NULL,
   TIITEM_NO             NUMBER(10) NOT NULL,
   TIITEM_COUNT          NUMBER(10) NOT NULL,
   TIITEM_TYPE         VARCHAR2(01) NOT NULL,
   TIITEM_LENGTH         NUMBER(10) NOT NULL,
   COLUMN_TYPE           NUMBER(10) NOT NULL,
   COLUMN_LENGTH         NUMBER(10) NOT NULL,
   TIMESTAMP           VARCHAR2(14) NOT NULL
   );

CREATE TABLE OT.TISET_KEY (
   TIBASE_NAME         VARCHAR2(26) NOT NULL,
   TISET_NO              NUMBER(10) NOT NULL,
   TIKEY_SEQ_NO          NUMBER(10) NOT NULL,
   GEN_PATHCOLUMN      VARCHAR2( 1) NOT NULL,
   TIMASTERSET_NO        NUMBER(10) NOT NULL,
   TIINDEX_ITEM_NO       NUMBER(10) NOT NULL,
   COLUMN_NAME        VARCHAR2(256) NOT NULL,
   TIINDEX_ITEM_NAME   VARCHAR2(20) NOT NULL,
   TISORT_ITEM_NO        NUMBER(10) NOT NULL,
   TIMESTAMP           VARCHAR2(14) NOT NULL
   );

/*************************************/
/* TIBASE:MUSIC.TIDATA.IMAXSOFT      */
/*************************************/

/*************************/
/* TISET:ALBUMS          */
/*************************/
INSERT INTO OT.TISET VALUES ('MUSIC.TIDATA.IMAXSOFT','ALBUMS','ALBUMS',1,'M',8,'20020716101418');

INSERT INTO OT.TISET_KEY VALUES
('MUSIC.TIDATA.IMAXSOFT',1,1,'N',1,1,'ALBUMCODE','ALBUMCODE',0,'20020716101418');

INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',1,1,'ALBUMCODE','ALBUMCODE',1,1,'I',2,1,10,'20020716101418');
INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',1,2,'ALBUMTITLE','ALBUMTITLE',3,1,'X',40,3,40,'20020716101418');
INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',1,3,'MEDIUM','MEDIUM',12,1,'X',2,3,2,'20020716101418');
INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',1,4,'ALBUMCOST','ALBUMCOST',2,1,'P',8,1,7,'20020716101418');
INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',1,5,'RECORDINGCO','RECORDINGCO',15,1,'X',16,3,16,'20020716101418');
INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',1,6,'DATERECORDED','DATERECORDED',9,1,'X',16,3,16,'20020716101418');
INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',1,7,'MFGCODE','MFGCODE',13,1,'X',40,3,40,'20020716101418');
INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',1,8,'COMMENT_IMS','COMMENT',7,1,'X',80,3,80,'20020716101418');
```

```
/**************************/
/* TISET:COMPOSERS        */
/**************************/
INSERT INTO OT.TISET VALUES
('MUSIC.TIDATA.IMAXSOFT','COMPOSERS','COMPOSERS',2,'M',4,'20020716101418');

INSERT INTO OT.TISET_KEY VALUES
('MUSIC.TIDATA.IMAXSOFT',2,1,'N',2,8,'COMPOSERNAME','COMPOSERNAME',0,'20020716101418');

INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',2,1,'COMPOSERNAME','COMPOSERNAME',8,1,'X',16,3,16,'20020716101418');
INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',2,2,'BIRTH','BIRTH',5,1,'X',16,3,16,'20020716101418');
INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',2,3,'DEATH','DEATH',10,1,'X',16,3,16,'20020716101418');
INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',2,4,'BIRTHPLACE','BIRTHPLACE',6,1,'X',40,3,40,'20020716101418');
INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',2,5,'COMMENT_IMS','COMMENT',7,1,'X',80,3,80,'20020716101418');

/**************************/
/* TISET:SELECTIONS-A     */
/**************************/
INSERT INTO OT.TISET VALUES ('MUSIC.TIDATA.IMAXSOFT','SELECTIONS_A','SELECTIONS-
A',3,'A',8,'20020716101418');

INSERT INTO OT.TISET_KEY VALUES
('MUSIC.TIDATA.IMAXSOFT',3,1,'N',3,16,'SELECTIONNAME','SELECTIONNAME',0,'20020716101418');

INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',3,1,'SELECTIONNAME','SELECTIONNAME',16,1,'X',40,3,40,'20020716101418');

/**************************/
/* TISET:SELECTIONS       */
/**************************/
INSERT INTO OT.TISET VALUES
('MUSIC.TIDATA.IMAXSOFT','SELECTIONS','SELECTIONS',4,'D',8,'20020716101418');

INSERT INTO OT.TISET_KEY VALUES
('MUSIC.TIDATA.IMAXSOFT',4,1,'Y',1,1,'ALBUMCODE','ALBUMCODE',0,'20020716101418');
INSERT INTO OT.TISET_KEY VALUES
('MUSIC.TIDATA.IMAXSOFT',4,2,'Y',3,16,'SELECTIONNAME','SELECTIONNAME',0,'20020716101418');
INSERT INTO OT.TISET_KEY VALUES
('MUSIC.TIDATA.IMAXSOFT',4,3,'Y',2,8,'COMPOSERNAME','COMPOSERNAME',0,'20020716101418');

INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',4,1,'ALBUMCODE','ALBUMCODE',1,1,'I',2,1,10,'20020716101418');
INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',4,2,'SELECTIONNAME','SELECTIONNAME',16,1,'X',40,3,40,'20020716101418');
INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',4,3,'COMPOSERNAME','COMPOSERNAME',8,1,'X',16,3,16,'20020716101418');
INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',4,4,'TIMING','TIMING',18,1,'X',16,3,16,'20020716101418');
INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',4,5,'PERFORMERS','PERFORMERS',14,1,'X',40,3,40,'20020716101418');
INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',4,6,'COMMENT_IMS','COMMENT',7,1,'X',80,3,80,'20020716101418');

/**************************/
/* TISET:LOG              */
/**************************/
INSERT INTO OT.TISET VALUES ('MUSIC.TIDATA.IMAXSOFT','LOG','LOG',5,'D',6,'20020716101418');

INSERT INTO OT.TISET_KEY VALUES
('MUSIC.TIDATA.IMAXSOFT',5,1,'Y',1,1,'ALBUMCODE','ALBUMCODE',0,'20020716101418');
INSERT INTO OT.TISET_KEY VALUES
('MUSIC.TIDATA.IMAXSOFT',5,2,'Y',3,16,'SELECTIONNAME','SELECTIONNAME',0,'20020716101418');

INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',5,1,'ALBUMCODE','ALBUMCODE',1,1,'I',2,1,10,'20020716101418');
```

```
INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',5,2,'SELECTIONNAME','SELECTIONNAME',16,1,'X',40,3,40,'20020716101418');
INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',5,3,'STARTTIME','STARTTIME',17,1,'X',16,3,16,'20020716101418');
INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',5,4,'ENDTIME','ENDTIME',11,1,'X',16,3,16,'20020716101418');
INSERT INTO OT.TISET_FIELD VALUES
('MUSIC.TIDATA.IMAXSOFT',5,5,'ANNOUNCER','ANNOUNCER',4,1,'X',40,3,40,'20020716101418');

/*************************************/
/* TIBASE:MUSIC_TIDATA_IMAXSOFT      */
/*************************************/

CREATE TABLE MUSIC_TIDATA_IMAXSOFT.ALBUMS (
   ALBUMCODE                     NUMBER(  10) NOT NULL, /* I[2] */
   ALBUMTITLE                  VARCHAR2(  40) NOT NULL, /* X[40] */
   MEDIUM                      VARCHAR2(   2) NOT NULL, /* X[2] */
   ALBUMCOST                     NUMBER(   7) NOT NULL, /* P[8] */
   RECORDINGCO                 VARCHAR2(  16) NOT NULL, /* X[16] */
   DATERECORDED                VARCHAR2(  16) NOT NULL, /* X[16] */
   MFGCODE                     VARCHAR2(  40) NOT NULL, /* X[40] */
   COMMENT_IMS                 VARCHAR2(  80) NOT NULL, /* X[80] */
   IMAXSOFT13_SEQ_NO             NUMBER(  20) NOT NULL
   );
   /* 8 ROWS */

CREATE UNIQUE INDEX MUSIC_TIDATA_IMAXSOFT.I01_ALBUMS ON MUSIC_TIDATA_IMAXSOFT.ALBUMS (ALBUMCODE);

CREATE UNIQUE INDEX MUSIC_TIDATA_IMAXSOFT.I02_ALBUMS ON MUSIC_TIDATA_IMAXSOFT.ALBUMS
(IMAXSOFT13_SEQ_NO);

CREATE TABLE MUSIC_TIDATA_IMAXSOFT.COMPOSERS (
   COMPOSERNAME                VARCHAR2(  16) NOT NULL, /* X[16] */
   BIRTH                       VARCHAR2(  16) NOT NULL, /* X[16] */
   DEATH                       VARCHAR2(  16) NOT NULL, /* X[16] */
   BIRTHPLACE                  VARCHAR2(  40) NOT NULL, /* X[40] */
   COMMENT_IMS                 VARCHAR2(  80) NOT NULL, /* X[80] */
   IMAXSOFT13_SEQ_NO             NUMBER(  20) NOT NULL
   );
   /* 4 ROWS */

CREATE UNIQUE INDEX MUSIC_TIDATA_IMAXSOFT.I01_COMPOSERS ON MUSIC_TIDATA_IMAXSOFT.COMPOSERS
(COMPOSERNAME);

CREATE UNIQUE INDEX MUSIC_TIDATA_IMAXSOFT.I02_COMPOSERS ON MUSIC_TIDATA_IMAXSOFT.COMPOSERS
(IMAXSOFT13_SEQ_NO);

CREATE TABLE MUSIC_TIDATA_IMAXSOFT.SELECTIONS_A (
   SELECTIONNAME               VARCHAR2(  40) NOT NULL, /* X[40] */
   IMAXSOFT13_SEQ_NO             NUMBER(  20) NOT NULL
   );
   /* 8 ROWS */

CREATE UNIQUE INDEX MUSIC_TIDATA_IMAXSOFT.I01_SELECTIONS_A ON MUSIC_TIDATA_IMAXSOFT.SELECTIONS_A
(SELECTIONNAME);

CREATE UNIQUE INDEX MUSIC_TIDATA_IMAXSOFT.I02_SELECTIONS_A ON MUSIC_TIDATA_IMAXSOFT.SELECTIONS_A
(IMAXSOFT13_SEQ_NO);

CREATE TABLE MUSIC_TIDATA_IMAXSOFT.SELECTIONS (
   ALBUMCODE                     NUMBER(  10) NOT NULL, /* I[2] */
   SELECTIONNAME               VARCHAR2(  40) NOT NULL, /* X[40] */
   COMPOSERNAME                VARCHAR2(  16) NOT NULL, /* X[16] */
   TIMING                      VARCHAR2(  16) NOT NULL, /* X[16] */
   PERFORMERS                  VARCHAR2(  40) NOT NULL, /* X[40] */
   COMMENT_IMS                 VARCHAR2(  80) NOT NULL, /* X[80] */
   IMAXSOFT13_PATH_01            NUMBER(  10) NOT NULL,
   IMAXSOFT13_PATH_02            NUMBER(  10) NOT NULL,
   IMAXSOFT13_PATH_03            NUMBER(  10) NOT NULL,
   IMAXSOFT13_SEQ_NO             NUMBER(  20) NOT NULL
   );
   /* 8 ROWS */

CREATE UNIQUE INDEX MUSIC_TIDATA_IMAXSOFT.I01_SELECTIONS ON MUSIC_TIDATA_IMAXSOFT.SELECTIONS
(ALBUMCODE, IMAXSOFT13_PATH_01);

CREATE UNIQUE INDEX MUSIC_TIDATA_IMAXSOFT.I02_SELECTIONS ON MUSIC_TIDATA_IMAXSOFT.SELECTIONS
(SELECTIONNAME, IMAXSOFT13_PATH_02);

CREATE UNIQUE INDEX MUSIC_TIDATA_IMAXSOFT.I03_SELECTIONS ON MUSIC_TIDATA_IMAXSOFT.SELECTIONS
(COMPOSERNAME, IMAXSOFT13_PATH_03);

CREATE UNIQUE INDEX MUSIC_TIDATA_IMAXSOFT.I04_SELECTIONS ON MUSIC_TIDATA_IMAXSOFT.SELECTIONS
(IMAXSOFT13_SEQ_NO);

CREATE TABLE MUSIC_TIDATA_IMAXSOFT.LOG (
   ALBUMCODE                     NUMBER(  10) NOT NULL, /* I[2] */
   SELECTIONNAME               VARCHAR2(  40) NOT NULL, /* X[40] */
   STARTTIME                   VARCHAR2(  16) NOT NULL, /* X[16] */
   ENDTIME                     VARCHAR2(  16) NOT NULL, /* X[16] */
   ANNOUNCER                   VARCHAR2(  40) NOT NULL, /* X[40] */
   IMAXSOFT13_PATH_01            NUMBER(  10) NOT NULL,
   IMAXSOFT13_PATH_02            NUMBER(  10) NOT NULL,
   IMAXSOFT13_SEQ_NO             NUMBER(  20) NOT NULL
   );
   /* 6 ROWS */

CREATE UNIQUE INDEX MUSIC_TIDATA_IMAXSOFT.I01_LOG ON MUSIC_TIDATA_IMAXSOFT.LOG (ALBUMCODE,
IMAXSOFT13_PATH_01);

CREATE UNIQUE INDEX MUSIC_TIDATA_IMAXSOFT.I02_LOG ON MUSIC_TIDATA_IMAXSOFT.LOG (SELECTIONNAME,
IMAXSOFT13_PATH_02);

CREATE UNIQUE INDEX MUSIC_TIDATA_IMAXSOFT.I03_LOG ON MUSIC_TIDATA_IMAXSOFT.LOG
(IMAXSOFT13_SEQ_NO);

CREATE TABLE MUSIC_TIDATA_IMAXSOFT.DOOR_SYNC  (
   DBNAME           CHAR(30) NOT NULL,
   LOGRECNUM        INTEGER  NOT NULL,
   LOGFILENAME      CHAR(30) NOT NULL
   );

INSERT INTO MUSIC_TIDATA_IMAXSOFT.DOOR_SYNC VALUES ('-',0,' ');

COMMIT WORK;
EXIT;
```

otgenDDL

Given an owner and table, the utility will generates SQL scripts for creation and/or drop SQLPlus scripts from an Oracle database.

NOTE: All passwords must be entered in their encrypted form.

```
OPENTURBO otgenDDL <A.02.01> iMaxsoft Corp. Copyright 2003

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

  -u ORACLE Login User/Encrypted Password
  -x <COMMAND>
  -o Owner Name (wildcard % is allowed)
  -t Table Name (wildcard % is allowed)
  -s Output Script Filename
```

Valid –x commands include:

```
genCreateTABLE, genCreateINDEX, genCreateTRIGGER,
genCreateSEQ, genCreatePRIV, genCreateCONSTRAINT,
genDropTABLE, genDropINDEX, genDropTRIGGER,
genDropSEQ, genDropPRIV, genDropCONSTRAINT,
genALL,
```

Only one –x command can be used per invocation of otgenDDL. Command functionalities are indicated by the command name. For example –x `genCreateTABLE` will generate the table creation scripts.

Example

```
: otgenDDL -u ot/JO \
           -x genCreateTABLE \
           -o EMPLOYEE \
           -t @ \
           -s migration
```

This will generate the SQL CreateTABLE script for the all tables under the EMPLOYEE database owners.

otgenLOAD

Converts table load script file created into ORACLE sqlldr and sqlplus format. It adjusts steps for performance and adds security provisions. `otgenLOAD` will create four files: `OTLoadDB.EXECUTE`, `OTLoadDB.DROP`, `OTLoadDB.SQLLDR`, and `OTLoadDB.GEN`. Use `OTLoadDB.EXECUTE` script to run the load script.

NOTE:
1. All passwords must be entered in their encrypted form.
2. Use `NOCACHE` *in all cases, unless there is a performance degradation*

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

usage: otgenLOAD -u -d -t -c -x
       -u  ORACLE Login User/Password

       -d  TurboIMAGE Fully Qualified Database Name
           DBName.GROUP.ACCOUNT

       -t  A TABLE_NAME or @ for ALL_TABLES of the
           specified TurboIMAGE database

       -c  (Optional) CACHE or NOCACHE
           CACHE (default): ORACLE will get multiple sequence numbers per request
           NOCACHE: ORACLE will get single sequence number at a time

       -x  (Optional) ORACLE 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

```
:otgenLOAD -u ot/JO                 \
           -d MUSIC.TIDATA.IMAXSOFT \
           -t @                     \
           -c NOCACHE
```

otgenrpt

Generates database access plan report using the OPENTURBO Profiler Control File.

```
OPENTURBO<A.03.00> IMAXSOFT Corp. Copyright 2005.

otgenrpt: produce DB Access Plan report via .ctl file.
   -iCTLFile OPENTURBO Profiler Control File
```

Output report name will be `<CTLFile>.rpt`

Example

```
:otgenrpt –I SQLD02.ctl
```

otgenSQL

Generates embedded SQL for MF-COBOL to access ORACLE database environments. Code generated from iMaxsoft’s TIfile is for manual datasets and detail datasets only.

Syntax:	otgencpy -a -t –o -s –r
-a	SQL OBJECT GENERATION OPTIONS HOST – HOST Variables COMM - Common routines FIND - DBFIND SQL statement GET - DBGET generic SQL statement PUT - DBPUT SQL statement UPD - DBUPDATE SQL statement DEL - DBDELETE SQL statement
-tTIFILE	Fully qualified name of OPENTURBO TIFile
-oOutput	Output filename. Filenames of the form [name].cpy are reserved and are invalid inputs.
-sDataset	IMAGE Dataset Name. If Dataset is unset, the default behavior is to generate code for all datasets.
-s_COMMON_	*OPTIONAL:* Generate common variables only
-rOCCURS	*OPTIONAL:* SQL HOST Array Variables OCCURS, default 500 times

Examples

```
export TI="my_ti"
export DBSet="my_dbset"

#genCOMM
#Generates common routines (i.e. host connection, error reporting)
 export LTDBG1=0
 export LTDBG2=0
 export LTDBGOUT=-
 otgenSQL -a COMM -t /users/home/db/$TI.ti -o $DBSet -s $DBSet

#genHOSTset
# Creates file containing host variables
 export LTDBG1=0
 export LTDBG2=0
 export LTDBGOUT=-
 otgenSQL -a HOST -t /users/home/db/$TI.ti -o $DBSet -s $DBSet

otgenSQL -a HOST -t /users/home/db/consdbti.ti -o membermstr -s
membermstr

#genHOSTcom
# Builds Common Host Variable   File
 export LTDBG1=0
 export LTDBG2=0
 export LTDBGOUT=-
 otgenSQL -a HOST -t /users/home/db/$TI.ti -o common -s _COMMON_
```

```
#genPUTset
# Generates SQL for DBPUT
 export LTDBG1=0
 export LTDBG2=0
 export LTDBGOUT=-
 otgenSQL -a PUT -t /users/home/db/$TI.ti -o $DBSet -s $DBSet

#genUPDset
# Generates SQL for DBPUPDATE
 export LTDBG1=0
 export LTDBG2=0
 export LTDBGOUT=-
 otgenSQL -a UPD -t /users/home/db/$TI.ti -o $DBSet -s $DBSet

#genDELset
# Generates SQL for DBDELETE
 export LTDBG1=0
 export LTDBG2=0

#genFINDset
# Generates SQL for DBFIND
 export LTDBG2=0
 export LTDBGOUT=-
 otgenSQL -a FIND -t /users/home/db/$TI.ti -o $DBSet -s $DBSet

#genGETset
# Generates SQL for DBGET
 export LTDBG1=0
 export LTDBG1=0
 export LTDBG2=0
 export LTDBGOUT=-
 otgenSQL -a GET -t /users/home/db/$TI.ti -o $DBSet -s $DBSet
```

otINTEGRITY

otINTEGRITY scans and checks ORACLE database objects created by OPENTURBO at TurboIMAGE level to ensure the integrity of your ORACLE 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 ORACLE 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 ORACLE Login User/Encrypted Password
       -d TurboIMAGE Fully Qualified Database Name
          DBName.GROUP.ACCOUNT.
       -t TI File Name
       -r ORACLE RESERVE WORD File Name
       -p ORACLE Encrypted Password.for User
          DBName_GROUP_ACCOUNT.
```

otINTEGRITY will check the following areas:

Database Integrity

Database Structure

- Table: table name, table type (logical type, Automatic, Manual master, and Detailed), 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. ORACLE users and passwords (TurboIMAGE classes and passwords), and ORACLE user's database objects access authorities, objects are tables, columns, sequences, synonym (name domain), session, DBA, and DDL.

Database Checksums

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

otINTEGRITY views

You must create OPENTURBO system views first in order to use the `otINTEGRITY` views. For example:

```
$cd /opt/imaxsoft/OPENTURBO3.7/oracle/admin/view
$sqlplus
. . .
. . . <login in ot/ot>
. . .
SQL> @all_view.sql;
SQL> commit work;
SQL> exit;

// Script - all_view.sql:

@own_tbl
@tbl_col
@tbl_con
@tbl_con_col
@tbl_idx
@tbl_idx_col
@tbl_role
@tbl_role_dba
@tbl_sec
@tbl_sec_col
@tbl_seq
@tbl_synonyms
@tbl_system_priv
@tbl_trigger

// Script - drop_view.sql:

drop view OT.OT_TABLE;
drop view OT.OT_COLUMN;
drop view OT.OT_CONSTRAINT;
drop view OT.OT_CON_COLUMN;
drop view OT.OT_INDEX;
drop view OT.OT_IND_COLUMN;
drop view OT.OT_ROLE;
drop view OT.OT_ROLE_DBA;
drop view OT.OT_TBL_SECURITY;
drop view OT.OT_COL_SECURITY;
drop view OT.OT_SEQUENCE;
drop view OT.OT_SYNONYM;
drop view OT.OT_SYS_PRIVILEGE;
drop view OT.OT_TRIGGER;

// Script - own_tbl.sql:

rem View Name  :OT.OT_TABLE
rem Definition :To Get All The OPENTURBO Table Names
rem
```

```
rem Author     :Steven Tsung
rem Date       :10/23/2002
rem
rem Certified  :2002-11-06 Lee Tsai
rem
create or replace view OT.OT_TABLE as
select u.name AS OWNER, o.name AS TABLE_NAME
 from  sys.user$ u, sys.obj$ o, sys.tab$ t
where  o.owner# = u.user#
  and  o.obj#   = t.obj#
/

// Script - tbl_col.sql:

rem View Name  :OT.OT_COLUMN
rem Definition :To Get All The OPENTURBO Column Names
rem
rem Author     :Steven Tsung
rem Date       :10/23/2002
rem
create or replace view OT.OT_COLUMN as
select u.name AS OWNER_NAME, o.name AS TABLE_NAME, c.name AS COLUMN_NAME,
       decode(c.type#, 1, decode(c.charsetform, 2, 'NVARCHAR2', 'VARCHAR2'),
                       2, decode(c.scale, null,
                                 decode(c.precision#, null, 'NUMBER', 'FLOAT')
                                 , 'NUMBER'),
                       8, 'LONG', 9, decode(c.charsetform,
                                            2, 'NCHAR VARYING', 'VARCHAR'),
                       12, 'DATE', 23, 'RAW', 24, 'LONG RAW',
                       69, 'ROWID',
                       96, decode(c.charsetform, 2, 'NCHAR', 'CHAR'),
                       105, 'MLSLABEL',
                       106, 'MLSLABEL',
                       111, o.name,
                       112, decode(c.charsetform, 2, 'NCLOB', 'CLOB'),
                       113, 'BLOB', 114, 'BFILE', 115, 'CFILE',
                       121, o.name,
                       122, o.name,
                       123, o.name,
                       178, 'TIME(' ||c.scale|| ')',
                       179, 'TIME(' ||c.scale|| ')' || ' WITH TIME ZONE',
                       180, 'TIME(' ||c.scale|| ')',
                       181, 'TIME(' ||c.scale|| ')' || ' WITH TIME ZONE',
                       231, 'TIME(' ||c.scale|| ')' || ' WITH TIME ZONE',
                       182, 'INTERVAL YEAR(' ||c.precision#|| ') TO MONTH',
                       183, 'INTERVAL DAY(' ||c.precision#|| ') TO SECOND(' ||
                             c.scale || ')',
                       208, 'UROWID',
                       'UNDEFINED')  AS DATA_TYPE,
       c.length AS DATA_LENGTH, c.precision# AS DATA_PRECISION,
       c.scale  AS DATA_SCALE,
       decode(sign(c.null$), -1, 'D', 0, 'NULL', 'NOT NULL') AS ISNULL,
       c.col# AS COLUMN_ID
  from sys.col$ c, sys.user$ u, sys.obj$ o
  where o.owner# = u.user#
    and o.obj#   = c.obj#
/

//Script - tbl_con.sql:

rem View Name  :OT.OT_CONSTRAINT
rem Definition :To Get All The OPENTURBO Constraint Names
rem
rem Author     :Steven Tsung
rem Date       :10/23/2002
rem
create or replace view OT.OT_CONSTRAINT as
select ou.name as OWNER          ,oc.name as CONSTRAINT_NAME,
       decode(c.type#, 1, 'C', 2, 'P', 3, 'U',
              4, 'R', 5, 'V', 6, 'O', 7 , 'C', '?') as C_TYPE,
       o.name  as TABLE_NAME        ,ru.name  as R_OWNER,
       rc.name as R_CONSTRAINT_NAME
  from sys.con$ oc, sys.con$ rc, sys.user$ ou, sys.user$ ru,
       sys.obj$ o,  sys.cdef$ c
 where oc.owner# = ou.user#
   and oc.con#   = c.con#
   and c.obj#    = o.obj#
   and c.type#  != 8
   and c.rcon#   = rc.con#(+)
   and rc.owner# = ru.user#(+)
/

//Script - tbl_con_col.sql:

rem View Name  :OT.OT_CON_COLUMN
rem Definition :To Get All The OPENTURBO Constrain Column Names
rem
rem Author     :Steven Tsung
rem Date       :10/23/2002
rem
create or replace view OT.OT_CON_COLUMN as
select u.name as OWNER           ,c.name CONSTRAINT_NAME,
       o.name as TABLE_NAME,
       decode(bitand(col.property, 524289), 1, ac.name, 524288, ac.name,
              col.name) as COLUMN_NAME, cc.pos# as POSITION
  from sys.user$ u, sys.con$ c, sys.col$ col, sys.ccol$ cc, sys.cdef$ cd,
       sys.obj$ o,  sys.attrcol$ ac
 where c.owner#    = u.user#
   and c.con#      = cd.con#
   and cd.con#     = cc.con#
   and cc.obj#     = col.obj#
   and cc.intcol#  = col.intcol#
   and cc.obj#     = o.obj#
   and col.obj#    = ac.obj#(+)
   and col.intcol# = ac.intcol#(+)
/

//Script - tbl_idx.sql:

rem View Name  :OT.OT_INDEX
rem Definition :To Get All The OPENTURBO INDEX Names
rem
rem Author     :Steven Tsung
rem Date       :10/23/2002
rem
create or replace view OT.OT_INDEX as
select u.name  as OWNER       , o.name  as INDEX_NAME,
       iu.name as TABLE_OWNER , io.name as TABLE_NAME,
       decode(bitand(i.property, 16), 0, '', 'FUNCTION-BASED ') ||
        decode(i.type#, 1, 'NORMAL' ||
                           decode(bitand(i.property, 4), 0, '', 4, '/REV'),
                        2, 'BITMAP', 3, 'CLUSTER', 4, 'IOT - TOP',
                        5, 'IOT - NESTED', 6, 'SECONDARY', 7, 'ANSI', 8, 'LOB',
                        9, 'DOMAIN') as INDEX_TYPE
  from sys.user$ u, sys.obj$ o, sys.user$ iu, sys.obj$ io, sys.ind$ i
 where o.owner#  = u.user#
   and o.obj#    = i.obj#
   and i.bo#     = io.obj#
   and io.owner# = iu.user#
/

//Script - tbl_idx_col.sql:

rem View Name  :OT.OT_IND_COLUMN
rem Definition :To Get All The OPENTURBO Index Column Names
rem
rem Author     :Steven Tsung
rem Date       :10/23/2002
rem
create or replace view OT.OT_IND_COLUMN as
select io.name as OWNER       , idx.name  as INDEX_NAME,
       decode(bitand(i.property, 16), 0, '', 'FUNCTION-BASED ') ||
        decode(i.type#, 1, 'NORMAL' ||
                           decode(bitand(i.property, 4), 0, '', 4, '/REV'),
                        2, 'BITMAP', 3, 'CLUSTER', 4, 'IOT - TOP',
                        5, 'IOT - NESTED', 6, 'SECONDARY', 7, 'ANSI',
                        8, 'LOB',
                        9, 'DOMAIN') INDEX_TYPE,
       bo.name as TABLE_OWNER , base.name as TABLE_NAME,
       decode(bitand(c.property, 524289), 1, ac.name, 524288, ac.name,
                     c.name) as COLUMN_NAME,
       decode(bitand(c.property, 131072), 131072, 'DESC', 'ASC') as COL_ORDER
  from sys.col$ c,   sys.obj$ idx, sys.obj$ base, sys.icol$ ic,
       sys.user$ io, sys.user$ bo, sys.ind$ i,    sys.attrcol$ ac
  where ic.bo#     = c.obj#
    and ic.intcol# = c.intcol#
    and ic.bo#     = base.obj#
    and io.user#   = idx.owner#
    and bo.user#   = base.owner#
    and ic.obj#    = idx.obj#
    and idx.obj#   = i.obj#
    and i.type# in (1, 2, 3, 4, 6, 7, 9)
    and c.obj#     = ac.obj#(+)
    and c.intcol#  = ac.intcol#(+)
/

//Script - tbl_role.sql:

rem View Name  :OT.OT_ROLE
rem Definition :To Get All The OPENTURBO Role Names
rem
rem Author     :Steven Tsung
rem Date       :10/23/2002
rem
create or replace view OT.OT_ROLE as
select decode(sa.grantee#, 1, 'PUBLIC', u1.name) as GRANTEE,
       u2.name  as GRANTED_ROLE
  from sys.sysauth$ sa, sys.user$ u1, sys.user$ u2, sys.defrole$ ud
  where sa.grantee#     =ud.user#(+)
    and sa.privilege#   =ud.role#(+) and u1.user#=sa.grantee#
    and u2.user#        =sa.privilege#
  group by decode(sa.grantee#,1,'PUBLIC',u1.name), u2.name
/

//Script - tbl_role_dba.sql:

rem View Name  :OT.OT_ROLE_DBA
rem Definition :To Get All The DBA Role Names
rem
rem Author     :Steven Tsung
rem Date       :10/23/2002
rem
create or replace view OT.OT_ROLE_DBA as
select decode(sa.grantee#, 1, 'PUBLIC', u1.name) as GRANTEE,
       u2.name as GRANTED_ROLE
  from sys.sysauth$ sa, sys.user$ u1, sys.user$ u2, sys.defrole$ ud
  where sa.grantee#     =ud.user#(+)
    and sa.privilege#   =ud.role#(+) and u1.user#=sa.grantee#
    and u2.user#        =sa.privilege#
    and u2.name         ='DBA'
  group by decode(sa.grantee#,1,'PUBLIC',u1.name), u2.name
/

//Script - tbl_sec.sql:

rem View Name  :OT.OT_TBL_SECURITY
rem Definition :To Get All The OPENTURBO Table Granted Security
rem
rem Author     :Steven Tsung
rem Date       :10/23/2002
rem
create or replace view ot.OT_TBL_SECURITY as
select ue.name  as GRANTEE    , u.name  as OWNER,
       o.name   as TABLE_NAME , ur.name as GRANTOR,
       tpm.name as PRIVILEGE
  from sys.objauth$ oa, sys.obj$ o, sys.user$ u, sys.user$ ur,  sys.user$ ue,
       table_privilege_map tpm
 where oa.obj#       = o.obj#
   and oa.grantor#   = ur.user#
   and oa.grantee#   = ue.user#
   and oa.col# is null
   and oa.privilege# = tpm.privilege
   and u.user#       = o.owner#
/

//Script - tbl_sec_col.sql:

rem View Name  :OT.OT_COL_SECURITY
rem Definition :To Get All The OPENTURBO Column Granted Security
rem
rem Author     :Steven Tsung
rem Date       :10/23/2002
rem
create or replace view OT.OT_COL_SECURITY as
select ue.name  as GRANTEE    , u.name  as OWNER,
       o.name   as TABLE_NAME , c.name  as COLUMN_NAME,
       ur.name  as GRANTOR    ,
       tpm.name as PRIVILEGE
  from sys.objauth$ oa, sys.obj$ o, sys.user$ u, sys.user$ ur,  sys.user$ ue,
       sys.col$ c,      table_privilege_map tpm
 where oa.obj#       = o.obj#
   and oa.grantor#   = ur.user#
   and oa.grantee#   = ue.user#
   and oa.obj#       = c.obj#
   and oa.col#       = c.col#
   and bitand(c.property, 32) = 0
   and oa.col# is null
   and oa.privilege# = tpm.privilege
   and u.user#       = o.owner#
/

//Script - tbl_seq.sql:

rem View Name  :OT.OT_SEQUENCE
rem Definition :To Get All The OPENTURBO Sequence Names
rem
rem Author     :Steven Tsung
rem Date       :10/23/2002
rem
rem Certified  :2002-11-06 Lee Tsai
rem
create or replace view OT.OT_SEQUENCE as
select u.name       as OWNER  , o.name      as SEQUENCE_NAME,
       s.minvalue   as MIN    , s.maxvalue  as MAX,
       s.increment$ as INC    ,
       decode(s.cycle#, 0, 'N', 1, 'Y')     as CYCLE,
       decode(s.order$, 0, 'N', 1, 'Y')     as ORDER_NO,
       s.cache      as ISCACHE, s.highwater as LAST_NUMBER
  from sys.seq$ s, sys.obj$ o, sys.user$ u
  where u.user#  = o.owner#
    and o.obj#   = s.obj#
/

//Script - tbl_synonyms.sql:

rem View Name  :OT.OT_SYNONYM
rem Definition :To Get All The OPENTURBO Synonym Names
rem
rem Author     :Steven Tsung
rem Date       :10/23/2002
rem
create or replace view OT.OT_SYNONYM as
select u.name  as OWNER      , o.name  as SYNONYM_NAME,
       s.owner as TABLE_OWNER, s.name  as TABLE_NAME
  from sys.user$ u, sys.syn$ s, sys.obj$ o
 where o.obj#   = s.obj#
   and o.type#  = 5
   and o.owner# = u.user#
/

//Script - tbl_system_priv.sql:

rem View Name  :OT.OT_SYS_PRIVILEGE
rem Definition :To Get All The OPENTURBO System Privilege
rem
rem Author     :Steven Tsung
rem Date       :10/23/2002
rem
create or replace view ot.ot_sys_privilege as
select u.name as GRANTEE  , spm.name as PRIVILEGE
  from sys.system_privilege_map spm, sys.sysauth$ sa, sys.user$ u
 where sa.grantee#   = u.user#
   and sa.privilege# = spm.privilege
/

//Script - tbl_trigger.sql:

rem View Name  :OT.OT_TRIGGER
rem Definition :To Get All The OPENTURBO Trigger Names
rem
rem Author     :Steven Tsung
rem Date       :10/23/2002
rem
create or replace view OT.OT_TRIGGER as
select trigusr.name as OWNER      , trigobj.name as TRIGGER_NAME,
       tabusr.name  as TABLE_OWNER, tabobj.name  as TABLE_NAME,
       decode(t.insert$*100 + t.update$*10 + t.delete$,
              100, 'INSERT',
              010, 'UPDATE',
              001, 'DELETE',
              110, 'INSERT OR UPDATE',
              101, 'INSERT OR DELETE',
              011, 'UPDATE OR DELETE',
              111, 'INSERT OR UPDATE OR DELETE', 'ERROR') as TRIGGER_EVENT,
       t.definition as TRIGGER_CONTENT
  from sys.obj$  trigobj, sys.obj$  tabobj, sys.trigger$ t,
       sys.user$ tabusr,  sys.user$ trigusr
 where trigobj.obj#   = t.obj#
   and tabobj.obj#    = t.baseobject
   and tabobj.owner#  = tabusr.user#
   and trigobj.owner# = trigusr.user#
/
```

Running otINTEGRITY:

An example of running `otINTEGRITY` is shown below:

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

This will give you the following summary report:

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

```
ORACLE_USER      - [OT/JO]
TIBASE_NAME      - [MUSIC.TIDATA.IMAXSOFT] [MUSIC_TIDATA_IMAXSOFT]
TIFile           - [/opt/imaxsoft/OPENTURBO3.7/oracle/musicdemo/tidb/ti]
RWFile           - [/opt/imaxsoft/OPENTURBO3.7/oracle/conf/RESERVE.ORACLE]
OWNER_PASSWD     - [HPNDX]

----- ORACLE 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]

----- ORACLE 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 ORACLE_USER = [OT/JO] >>
<< otCHECKChecksum TIBASE_NAME = [MUSIC.TIDATA.IMAXSOFT] >>
<< otCHECKChecksum TIFile      = [/opt/imaxsoft/OPENTURBO3.7/oracle/musicdemo/tidb/ti]
>>

Checksum Summary Report:

----------- ------- FROM DATABASE -------- --------- FROM TIFile --------
DBNAME    : MUSIC.TIDATA.IMAXSOFT          MUSIC.TIDATA.IMAXSOFT
TIFILE    : ../tidb/ti
/opt/imaxsoft/OPENTURBO3.7/oracle/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
```

otSETChecksum

Sets the checksum into ORACLE database and to ensure that your ORACLE database structure is in sync with the OPENTURBO TIFile. *You must run this program to sync ORACLE and TIFile, whenever there is a change to TIFile.*

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

usage: otSETChecksum -u -d -t -v
       -u ORACLE Login User/Password
       -d TurboIMAGE Fully Qualified Database Name
          DBName.GROUP.ACCOUNT.
       -t TIFile Name
       -v View Checksum Summary
```

Example

```
: otSETChecksum -u OT/JO            \
              -d MUSIC.DATA.TICHG \
              -t MUSIC.DATA.TICHG \
              -v
```

This will return

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

ORACLE_USER  - [OT/JO]
TIBASE_NAME  - [MUSIC.DATA.TICHG]
TIFile       - [MUSIC.DATA.TICHG]
View         - [1]

 ====================== ========================== ==========================
                        TIBASE                     TIFile
 ---------------------- -------------------------- --------------------------
 TIBASE_NAME            MUSIC.DATA.TICHG           MUSIC.DATA.TICHG
 TIFILE_NAME            MUSIC.DATA.TICHG           MUSIC.DATA.TICHG
 TIFILE_VERSION         A.01.03                    A.01.03
 CHECKSUM_ITEMNO        253                        253
 CHECKSUM_SETNO         518                        518
 CHECKSUM_PATHNO        204                        204
 TIMESTAMP              20020910105145             20020910105145
```

otsyncSEQ

Scans and checks all datasets of a specific or all databases in OT.TISET and OT.TIBASE for the integrity of next available number of SEQUENCE object versus the maximum value of IMAXSOFT13_SEQ_NO. This program detects, reports and fixes any SEQUENCE that is equal to or smaller than the maximum value of IMAXSOFT13_SEQ_NO. This program also creates new SEQUENCE objects and their associated ACCESS AUTHORITIES, if they are missing.

```
otsyncSEQ <A.03.00> IMAXSOFT Corp. Copyright 2004.
usage: otsyncSEQ -d -u -r -v -x
       otsyncSEQ processes tables in OT.TISET for all or the
       specific TurboIMAGE database(s) and uses TIFILE_NAME
       in OT.TIBASE, so always keep OT tables up-to-date.
       This program will validate and reset table SEQUENCE to
       MAX(IMAXSOFT13_SEQ_NO)+1; create new SEQUENCE and its
       ACCESS AUTHORITIES for all users, if it doesn't exist.
   -d  TurboIMAGE Database Name; skip -d option means for all
       TurboIMAGE databases.
   -u  ORACLE Login User/Password
   -r  ORACLE RESERVE WORD File Name
   -v  Validate the next available SEQUENCE vs MAX(IMAXSOFT13_SEQ_NO)
       only, no re-setting of SEQUENCE.
   -x  Reset SEQUENCE to MAX(IMAXSOFT13_SEQ_NO)+1
```

recover, reader

OPENTURBO provides a set of recovery tools for disaster recovery. At the core is the `recover` utility that will read a cache log file and roll back the migration back to the original source database. Any additional transactions that have occurred since the original migration point will be applied to the recovered database.

```
OPENTURBO recover<A.03.07> iMaxsoft Corp. Copyright 2005

usage: prerecov LogfileName Host Port Pgm DBpwdFile
       LogfileName - * for all files from current dir
       DBpwdFile record format - DATABASE,PASSWORD,RealBASE
     * RealBASE is the actual server DBname:
       HP3000 recover to HP9000, it is the Eloquence DBname;
       HP9000 recover to HP3000, it is the IMAGE DBname
```

OPENTURBO also provides a utility, `reader`, to read the Log Files used for disaster recovery.

```
OPENTURBO reader<A.03.07> iMaxsoft Corp. Copyright 2005
usage: reader LogfileName
```

tidrv

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 ORACLE 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 `$_OTB_ROOT/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 ORACLE 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 ORACLE database
- Running `TIDRV -turboimage` with `libot.sl` accesses remote TurboIMAGE database on HP e3000 via OPENTURBO HP-UX library

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/oracle/tmp/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   =>+00000000000000089
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=-
$/opt/imaxsoft/OPENTURBO3.7/oracle/tidrv –iintrx1 –oouttrx1
```

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

ticopy

Makes a copy of OPENTURBO TIFILE with a different TurboIMAGE name or to upgrade/migrate an older TIFILE to the `TICOPY` version.

```
OPENTURBO TICOPY <A.01.03> iMaxsoft Corp. Copyright 2002.
run ticopy.bin;info='-d -t -r -n -o'
   -dDBName : CopyFrom Fully Qualified TurboIMAGE DBName
   -tTIFILE : CopyFrom Fully Qualified TIFILE Name
   -rRWFile : Reserve Word File Name
   -nDBName : CopyTo Fully Qualified TurboIMAGE DBName
              Ignore means NO Change of TurboIMAGE DBName
   -oTIFILE : CopyTo Fully Qualified TIFILE Name
              Ignore means TIFILE Version Migration
```

Examples

```
// TIFILE Duplication

ticopy.bin;info=’-dMUSIC.TIDATA.IMAXSOFT -tti.otcmd -rreserve.oracle -
nMUSIC.TEST.IMAXSOFT -oMUSICTI.otcmd’

// TIFILE Version Upgrade

ticopy.bin;info=’-dMUSIC.TIDATA.IMAXSOFT -tti.otcmd -rreserve.oracle’
```

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 '
```

Option	Description
-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.ORACLE.IMAX'
```

2. Comparing Version of TI File to IMAGE DB

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

3. Outputting IMAGE DB to File

```
tiloadam.bin.imaxsoft;info='-tINVENTTI –mOUT1 -rRESERVE.ORACLE.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 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: 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 B: 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

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