TurboIMAGE to Eloquence Migration

Michael Marxmeier, Marxmeier Software AG
info@marxmeier.com

Session 3063

Overview

- Eloquence overview
- Introduction to the Eloquence database architecture
- Installation / Getting started / Administration
- TurboIMAGE migration, possible problems and solutions
- Future development
- Live demonstration and tutorial

Eloquence at a glance

- Excellent compatibility and performance for IMAGE based applications
- Cost effective
- Supports multiple platforms
- Proven solution

Excellent compatibility

- All TurboIMAGE intrinsics are supported and behave (almost) identical
- HP3000 applications can typically be ported with no or only minor changes
- Eloquence is supported by a range of former HP3000 utilities

Cost effective

- Eloquence saves considerable time and effort in the migration process and allows focusing on other tasks
- Eloquence is easy to manage and retains existing know how
- Eloquence is priced attractively

Complete package

- The Eloquence database comes with
  - Comprehensive set of database utilities
  - Structural maintenance
  - Integrated indexing (TPI subset)
  - On-line backup
  - MPE migration tools

Product history

- Eloquence was created by Marxmeier Software and sold to Hewlett-Packard
- Eloquence was first released in 1989 as a migration solution to move HP250/HP260 applications to HP-UX

Product history

- Marxmeier Software has been responsible for Eloquence development and support
- The Eloquence product was transferred back to Marxmeier Software AG in 2002

Product components

- Eloquence programming language (based on HP Business Basic)
- Eloquence database (based on IMAGE)
- Graphical User Interface
- Development environment

Product overview

- About 2500+ installations worldwide
- Used by about 60+ VARs / ISVs worldwide
- Covers a wide range of industries and sizes from single user to a few hundred concurrent users

eloquence

Eloquence Database Architecture

An introduction to the Eloquence database architecture

Overview

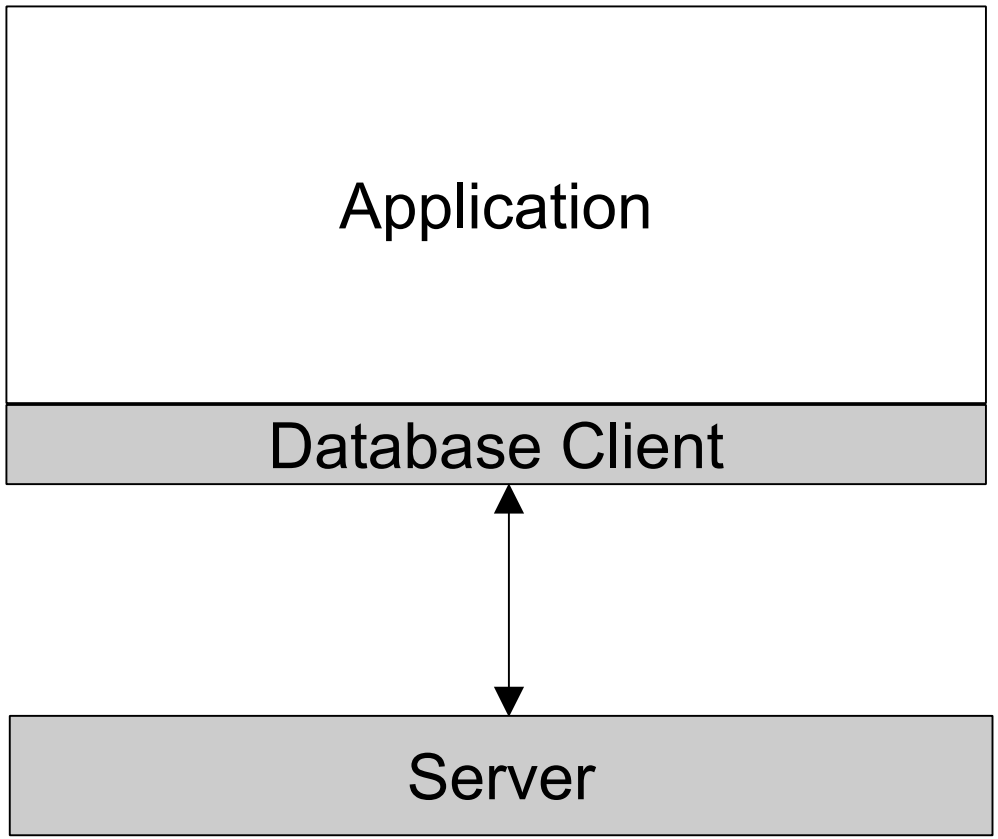
- The Eloquence database is almost 100% compatible to TurboIMAGE at the application level
- The underlying architecture is different

eloquence

Client/Server architecture

- Database access is performed by a server process
- The application is linked with the database API
- The server is connected through the network (or shared memory)

Client-server architecture



Network transparent

- Applications running on different machines and operating systems can access a common database
- Requests and results are translated transparently
  - Character set encoding
  - Byte order conversion

Multiple platforms

- Eloquence is available for multiple operating systems and architectures
  - HP-UX on PA-RISC and Itanium
  - Linux on Intel IA-32 (Itanium)
  - Windows NT/2000/XP on Intel IA-32
  - Database client library on MPE

Indexing

- Eloquence comes with integrated indexing
- Indexes are used instead of hashing with master sets
- Eloquence implements a commonly used subset of the TPI functionality

Locking

- Locking is fully compatible with TurboIMAGE
- Eloquence does not impose a locking strategy
  - Write operations do not require a previous lock. If a conflicting lock is granted, a status is returned

Locking (cont)

- READ Locks are supported
- Selective DBUNLOCK
- Multiple DBLOCKs are allowed
  - Deadlock conditions are detected and a status is returned

eloquence

Transactions

- All databases are part of a transaction
- Uncommitted changes are not visible to other processes
- Transactions are not limited in size
- Nested transactions

eloquence

Database names

- A database name is not restricted to 6 characters
- Databases do not reside in the file system but are managed through a server process
- A database name addresses a specific server instance instead of a file location

Database names

- Database name syntax

  `[[hostname][:service]/]database`

  – Hostname specifies database server system

  – Service specifies database server instance

Database names

- The following examples specify the same database:

```
localhost:eloqdb/SAMPLEDB
:eloqdb/SAMPLEDB
SAMPLEDB
```

Database names

- The EQ_DBSERVER environment variable may be used to specify the default server instance
- For example:

```
EQ_DBSERVER=invent9k.external.hp.com:8102
```

  - Specifies that the specified server instance manages the database.
  - The default is used unless a more specific information is provided

Database security

- The database server maintains a list of users
- Database access privileges are assigned to groups
  - Similar to IMAGE user classes
  - A user can be a member of multiple groups
  - IMAGE passwords are ignored

Database security

- The new DBLOGON procedure may be used to specify user and password
- A file can be specified to provide the user name or password
- A default user is used if no specific user is provided

Database security

- The EQ_DBUSER and EQ_DBPASSWORD environment variables may be used to specify the default user or the password

Database security

- For example:

```
EQ_DBUSER=file:/home/mike/dblogon
EQ_DBUSER=mike
EQ_DBPASSWORD=file:/home/mike/passwd
```

  - The default is used unless a more specific information is provided

Database environment

- A database environment consists of
  - a configuration file
  - one or more data volumes
  - a transaction log volume
- Multiple database environments can coexist on the same machine, each managed by a separate server process

Volume files

- Volume files are a storage container managed by the database server
- A maximum of 255 volume files are supported in a server environment
- The maximum size of a single volume file is 128 GB (currently limited to 2 GB on HP-UX and Linux)

Server catalog

- Eloquence does not use a ROOT file
- Structural information is maintained in the database environment
- The server catalog is initialized with the dbvolcreate utility and maintained with the schema and dbutil utilities

Database limits

- Eloquence B.07.00 Image limits
  - 2048 data items
  - 500 data sets
  - 64 / 16 paths
  - Entry length 5120 bytes

Scalability

- Database / data set size is limited by the disk space allocated to the database environment
  - Current limit is ~500 GB
  - Hard limit is ~32 TB
- Number of concurrent users per database environment is currently limited to 1000

eloquence

Database Utilities

An overview on the Eloquence database utilities

Offline utilities

- dbvolcreate / dbvolextend / dbvolchange / dblogreset - database volume management
- dbvoldump - display volume properties
- dbfsck - volume consistency check and simple repair tool
- dbcfix – database consistency check and repair tool
- dbrecover - forward recovery

eloquence

Administrative utilities

- dbctl - server management utility
- HTTP status monitor

Database utilities

- schema - Schema processor
- dbcreate / dberase / dbpurge - create / erase / purge database
- dbtables - database cross reference
- prschema - re-create schema from database
- dbdumpcat - catalog information utility

Database utilities

- dbexport / dbimport - export/import data base content to/from text file
- dbinfo - information on database tables
- dbutil - structural maintenance and database security management
- QUERY utility (different from QUERY/3000)

Installation and Configuration

Installation and Configuration of the Eloquence database

Overview

- Install the product and (OS or product) patches
- Configure the operating system
- Configure automatic server startup
- Create the database environment
- Platform differences

Evaluation license

- By default the “Personal Edition” license key is installed
- A temporary license key can be created during installation
- A temporary license key can be requested from the Eloquence web site

Create eloqdb user/group

- Create a user name and a group name (e.g. eloqdb) to be used as the owner/group of the database files
- On Windows the system account is used by default

Configure kernel parameters

- On Unix and Linux Eloquence can use shared memory for communication
- HP-UX kernel parameters need to be configured
  - semaphores related parameters
  - shared memory related parameters
  - process data size

Kernel parameters

- Semaphore configuration (EnableIPC enabled)
  - Set the **semmni** to at least *x*+20
  - Set the **semmap** to 'semmni' + 2
  - Set the **semmns** to at least *x*+*y*+20
  - Set the **semmnu** to at least *x*+20
  - Set the **semume** to at least *x*+20
    - x specifies the number of concurrent connections (Threads configuration item)
    - *y* specifies the number of i/o threads (IOThreads configuration item)

Kernel parameters

- Shared memory configuration (EnableIPC=1)
  - Set the **shmmni** to at least *x*+20
  - Set the **shmseg** to at least *x*+20
- Data size
  - Set the **maxdsiz** to at least 0x08000000 (128MB)

Setup database environment

- Database environment (server instance) consists of
  - Server configuration file (eloqdb.cfg)
  - Primary data volume
  - Transaction log volume(s)
  - Additional data volume(s) as required

Server configuration file

- The config file defines server properties
  - configuration
  - scaling and tuning parameters
  - volume files
- Default server configuration file is /etc/opt/eloquence6/eloqdb6.cfg

Simple server configuration

```
[Server]
 Service = eloqdb
 ServiceHTTP = 8103
 UID = eloqdb
 GID = eloqdb
 EnableIPC = 2
 SyncMode = 0
[Config]
 Threads = 100
 IOThreads = 4
 BufferCache = 64
 CheckPtSize = 50
```

Shared memory

- EnableIPC
  - EnableIPC=0 (default) disables use of shared memory communication
  - EnableIPC=1 enables use of shared memory on HP-UX and Linux
  - EnableIPC=2 enables use of a single shared memory segment for HP-UX (recommended)

Sync/Async mode

- SyncMode
  - SyncMode=1 (default) pushes all committed transactions to disk immediately and waits for completion
  - SyncMode=0 (recommended) writes changes to disk asynchroneously and does not wait for completion

Database server configuration

- **Threads**
  - Defines the max. number of concurrent connections for this server instance
- **IOThreads**
  - Defines the max. number of concurrent I/O operations (default=4)
  - Depends on the I/O capabilities
- **BufferCache**
  - Defines the memory reserved for the database cache

Create volume files

- dbvolcreate /var/opt/eloquence6/data01.vol
- dbvolextend –t log /var/opt/eloquence6/log.vol
- Optional: dbvolextend –t data /var/opt/eloquence6/data02.vol

Configure server startup

- Configure automatic startup of the Eloquence database
- The startup configuration file defines which Eloquence services are started
  - HP-UX: /etc/rc.config.d/eloquence6
  - Linux: /etc/sysconfig/eloquence6

Configure server startup

- The Eloquence **eloqsd** service is often not needed and should not be started
  - Set the **START_ELOQSD** variable to 0 to disable the automatic start of the eloqsd service

Start the database server

- HP-UX:
  - /sbin/init.d/eloq6 start|stop|status|restart [instance …]
- Linux:
  - /etc/init.d/eloq6 start|stop|status|restart [instance …]
- Operations:

start	– start server processes
stop	– stop server processes
status	– check status of server processes
restart	– restart server process

Troubleshooting

- The Eloquence database writes diagnostic messages to the syslog
  - HP-UX: /var/adm/syslog/syslog.log
  - Linux: /var/log/messages
  - Windows: application event log

Linux installation

- Eloquence uses the RPM package manager
  - RedHat Linux 7.x to 9 and SUSE Linux 7.x to 9 have been certified
  - Other Linux distributions may be used but additional manual configuration may be required

Linux installation

- For installation or update execute the command below

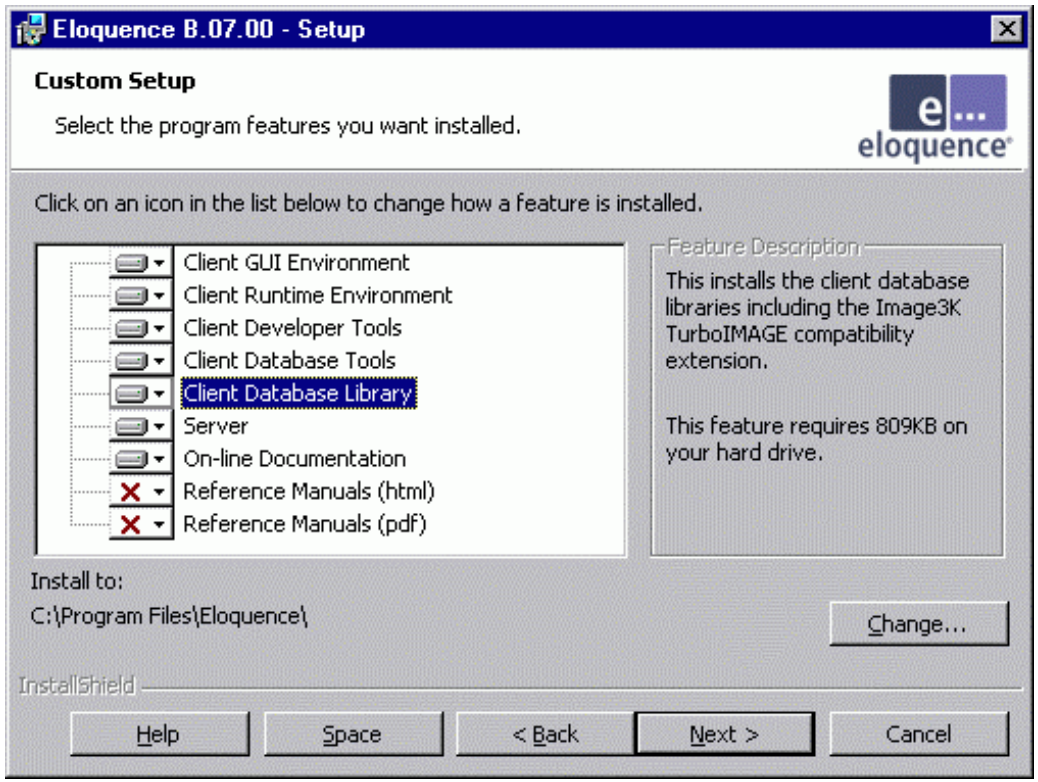
  $ rpm -U Eloquence-B0700.rh8.i386.rpm

- Temporary license option is not available during installation on Linux

Windows installation

- Eloquence uses the standard Windows Installer
- Different setup programs are used for Windows 2000/XP/2003, Windows NT and Windows 9x
- Different setup programs for download and CD-ROM installations

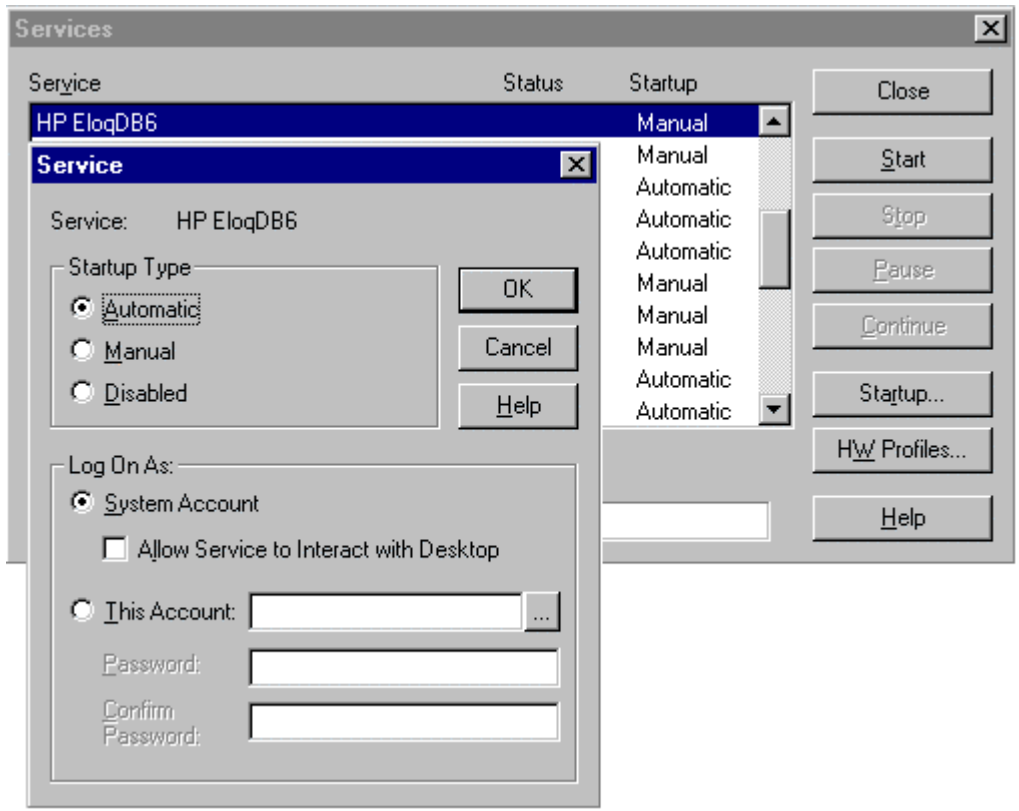
Select product features



Configure services

- Configure automatic start mode for the Eloquence database in the service control panel (eloqdb6 service)
- Start the eloqdb6 service manually for the first time
- The eloqsd service is often not needed and should not be started

Windows configuration automatic server start



eloquence

Administrative Procedures

Database backup

Database backup

- Supported backup strategies
  - Off-line backup
  - On-line backup
- Related options
  - Forward logging

Off-line backup

- Shutdown the eloqdb6 server process
- Backup all volume files
- Re-start the server process

On-line backup

- Enable on-line backup mode
- Backup the data volume file(s)
  - Backup of the log volume is optional
- Disable on-line backup mode

On-line backup

- In on-line backup mode, the data volumes are frozen
- Modifications during on-line backup are temporarily saved into the transaction log volume
- Any backup software can be used to create a consistent backup

On-line backup

- The dbctl utility is used to enable on-line backup mode
- Example backup script

```
$ dbctl -u file:/root/credentials backup start
$ tar -cf /dev/rmt/0m /database
$ dbctl -u file:/root/credentials backup stop
```

Forward logging

- Forward logging is used to record all modifications since a previous backup
- Forward logging is fast and involves only minimal processing
- The forward log files can be managed automatically by the server process

Forward logging

- Forward logging is enabled in the server configuration

```
[ForwardLog]
FwLog = /path/to/fwlog-%N.log
```

Database maintenance

- Make sure sufficient volume and disk space is available
  - Use the dbvoldump utility if the server is off-line
  - Use dbdumpcat or the HTTP status if the server is active

IMAGE Migration

How to migrate to Eloquence

TurboIMAGE compatibility

- All TurboIMAGE intrinsics are supported and behave (almost) identical
- HP e3000 applications can usually be ported with no or only minor changes
- Compatibility goes beyond intrinsic calls. Applications are built on assumptions and take advantage of specific behavior

TurboIMAGE compatibility

- Not supported:
  - DBCONTROL modes which are specific to TurboIMAGE implementation details
  - Item level security (all items considered writable)

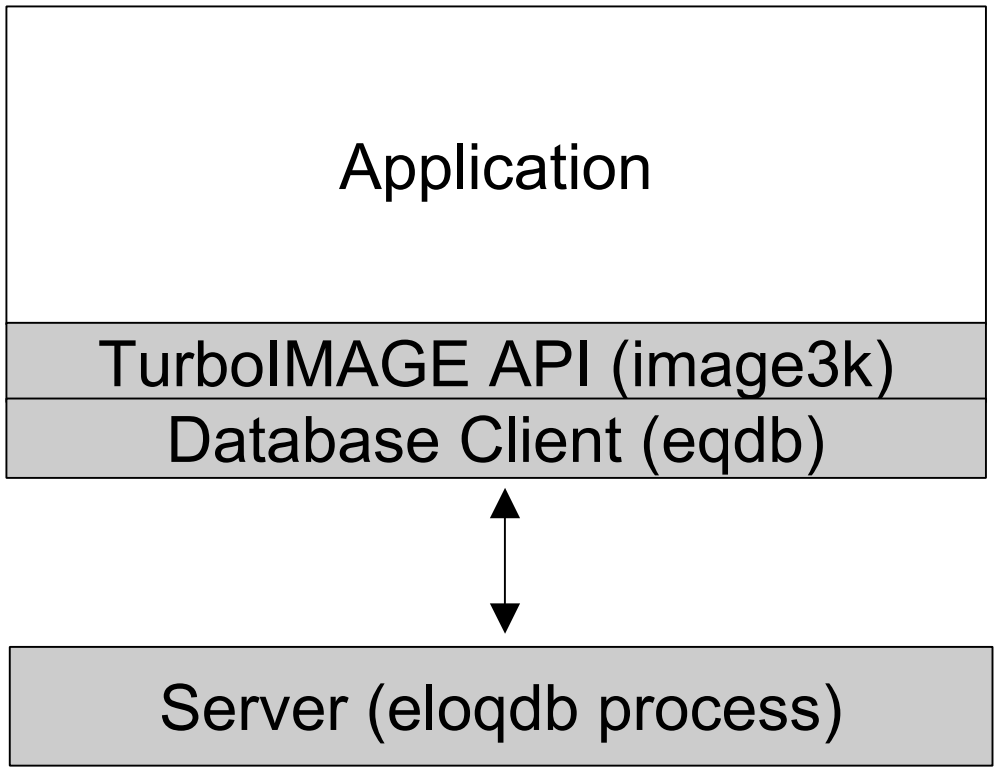
TurboIMAGE compatibility

- Required changes:
  - Eloquence requires the database name is terminated with a space, semicolon or NUL character
  - IMAGE passwords are ignored

TurboIMAGE compatibility

- TurboIMAGE compatibility is implemented at different levels
  - The database server implements functionality at the backend
  - The database client and utilities provide support for TurboIMAGE functionality
  - The TurboIMAGE compatibility API implements source code compatibility

TurboIMAGE compatibility



TurboIMAGE compatibility

- The Eloquence image3k library implements the TurboIMAGE intrinsics
- The application (or language runtime) is linked against the image3k library
- The image3k.h include file provides the function prototypes (C, C++)

Using Eloquence with AcuCOBOL

- Link the Eloquence image3k library to the ACU Cobol runtime (runcbl)
- Load the Eloquence image3k library dynamically (using CALL)

Using Eloquence with AcuCOBOL (cont)

- Eloquence currently uses native byte order
  - On little endian platforms (Intel IA-32) COMP-5 type must be used instead of COMP
  - The –D5 compiler option maps all COMP to COMP-5

Using Eloquence with MicroFocus Cobol

- Link the Eloquence image3k library to the application
- The SIGN “EBCDIC” compiler directive is required to make sure the binary encoding of ‘Z’ items is compatible

Using Eloquence with MicroFocus Cobol (cont)

- Eloquence currently uses native byte order
  - On little endian platforms (Intel IA-32) COMP-5 type must be used instead of COMP
  - A compiler directive may be used to map the COMP to the COMP-5 type

```
MAKESYN "COMP-5" = "COMP"
```

Migration Issues

Real World Issues

Data set capacity

- Data set capacity has a different meaning
  - Eloquence has no concept of a data set specific capacity
  - Eloquence returns the highest record number allocated for a data set as capacity value in DBINFO modes 202 and 205
- Eloquence data sets are dynamic and grow as required

Data set capacity

- Application may check for „enough room“ in a data set
- Solution:
  - Remove or disable capacity check
- Workaround:
  - Return „HUGE“ value as capacity
  - Trap Eloquence DBINFO 202 and 205 modes and return application specific „capacity“ value

Don’t lie to schema

- TurboIMAGE does not really care what you put in a character field
- Eloquence relies on type information
  - Eloquence may need to convert strings to different encoding
  - Eloquence may need to do a byte order conversion
  - Eloquence uses indexes which require type specific ordering

Don’t lie to schema

- Solution:
  - Use separate fields for different information
  - Use the correct item type
- Workaround:
  - Use Eloquence on a single platform
  - Use Eloquence binary item type ‘B’

Character set encoding

- On MPE the HP-ROMAN8 character set encoding is often used
  - HP-ROMAN8 encoding is typically not available on other platforms
  - Eloquence defaults to the HP-ROMAN8 character set on HP-UX (and MPE) and to ISO-8859-1 on other platforms
  - Eloquence performs conversion “on the fly”

Byte order

- PA-RISC and Itanium (with HP-UX) use big endian byte order
- Intel IA-32 and Itanium (Linux and Windows) use little endian byte order
- Eloquence performs conversion “on the fly” if necessary

Parameter alignment

- TurboIMAGE requires most arguments to be 16 bit aligned
- Eloquence relaxes most alignment restrictions
- Eloquence does not require a specific alignment for string arguments

Record numbers

- Eloquence uses a different algorithm to assign and re-use record numbers
  - TurboIMAGE uses a LIFO (last in first out) order to reuse deleted records (unless HWMPUT is active)
  - Eloquence uses a FIFO (first in first out) order to use available record numbers
  - Eloquence does not support HWPUT, application has no control over record number usage

Record numbers

- DBDELETE / DBPUT sequence likely results in different record number
- Solution:
  - Fix the application
- Workaround:
  - Use DBUPDATE mode 2 (same as DBUPDATE mode 1 and CIUPDATE)

Identical database names

- TurboIMAGE supports to use the same database name in different groups
- Eloquence requires an unique database name per server instance
- Solution:
  - Use multiple server instances (eg. test / production environments)
  - Add the group name to the database name (eg. DBNAME.GROUP)

Access to database files

- TurboIMAGE databases reside in the file system
- Applications could use file system operations to copy databases

Access to database files

- Eloquence databases reside in the volume files and are not accessible separately
- Solution
  - Copy whole database environment
  - Use dbstore to extract single database and dbrestore to restore database in another server instance
  - Use dbexport / dbimport

eloquence

Data Migration

Move your databases from
TurboIMAGE to Eloquence

Overview

- Schema files are compatible and no change is required
- Eloquence includes MPE tools to export the database content to flat files
- Transfer the schema file and the export files to the target system
- On the target system run the schema processor, the dbcreate utility and the dbimport utility

Export the database

- When running from the POSIX shell the arguments are separated by a space

```
$ DBEXPORT -p SECRET -v TESTDB
```

- When running from the MPE shell (CI) you need to enclose the arguments in quotes

```
: DBEXPORT "-p SECRET -v TESTDB"
```

Transfer the files

- Transfer your schema file and the export files to the Eloquence system
- When transferring by ftp
  - use text mode to transfer the schema file
  - use binary mode to transfer the export files

Create the database

- Run the Eloquence schema processor
  `$ dbschema schemafile`
  `$ schema -T schemafile`
  - Option -T selects TurboIMAGE compatibility mode
- Create the database
  `$ dbcreate database`

Import the data

- Use dbimport to load the database

```
$ dbimport -v -z roman8 database
```

  - The option -v displays the import progress
  - On the Windows and Linux platform you should specify the -z roman8 option to indicate the source data uses the HP-ROMAN8 encoding
  - This makes sure any national characters ("Umlaute") are converted

More information

- Detailed information is available on the Eloquence web site **http://www.hp-eloquence.com**
- Get in contact: **info@hp-eloquence.com**

Michael Marxmeier
mike@marxmeier.com