

Welcome.

The e3000 transition training program is proud to present: introduction to the Eloquence database.

Introduction to the Eloquence database will cover at a broad level the Eloquence database and TurboIMAGE compatibility.

© 2002 Marxmeier Software AG

The content of this document is subject to change without notice. Information is provided "as is" in the hope it is useful but any warranty on accuracy or usability is specifically disclaimed.

outline of contents

- Product overview
- Database concepts
- TurboIMAGE compatibility
- Database utilities
- Installation and configuration
- Data migration

During this course we will be starting with a short overview on the Eloquence product, its history and product structure

we then introduce the eloquence database architecture and explain the difference to TurboIMAGE

We cover the TurboIMAGE compatibility providing a detailed comparison at the intrinsic level

We provide an overview on the database utilities which come with eloquence

We then explain the installation and configuration of eloquence and finally we'll cover data migration, how to move your data from TurboIMAGE to Eloquence.

module 1 product overview

- Product overview
- Database concepts
- TurboIMAGE compatibility
- Database utilities
- Installation and configuration
- Data migration

In this module, we provide an short non technical overview on the Eloquence product.

product overview

- Product overview
- History
- Ordering
- Product structure
- Evaluation

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 4

The product overview covers the product structure, it's history, how it is used, ordering information, product evaluation and where to get more information.

Product Components

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

The Eloquence product consists of multiple components. Besides the database it comes with a complete software development and runtime environment and different options for user interfaces.

The Eloquence product includes the following major components:

- The Eloquence programming language, based on hp Business Basic
- The Eloquence database management system, based on IMAGE, that we are focussing on in this presentation
- Different user interface options (Text, Windows, Java and Web based user interfaces)
- Various development tools, including a graphical development environment on the Windows platform

Product Overview

- Eloquence is a product of Hewlett-Packard, Germany
- Support is available through the HP RC or from third party
- Eloquence is available on the HP-UX, Linux and Windows platform

Eloquence is a product of Hewlett-Packard, Germany.

Support is available through the hp response center or from third party

The Eloquence product is available for the HP-UX, Linux and Windows platform

Product Overview

About 2500+ installations worldwide

Used by about 60+ VARs / ISVs worldwide

Covers a wide range of installations

- Single user Laptop to 250+ concurrent users with 10+ GB database size
- Terminal based server solution to distributed GUI environment

Eloquence currently has about 2500 active installations worldwide.

It is used by about 60 VARs or ISVs worldwide that offer solutions based on Eloquence.

Eloquence installations cover a wide range

- from a single user laptop to a few hundred concurrent users and database sizes beyond 10 GB
- and terminal based server solutions to distributed GUI environments

Product Overview

Eloquence is typically used to implement vertical and custom specific solutions

Solutions based on Eloquence include

- ERP, Order Management
- Financial Accounting / Payroll
- Civil Services
- ...

Eloquence is typically used to implement vertical or custom specific solutions.

Eloquence based solutions include

- ERP, Order management
- Financial accounting, Payroll
- Civil services
- Bank, Financial services

Typically, Eloquence customers are small to medium sized companies which use Eloquence based solutions for business critical tasks, such as production planning, material management or order management.

**The HP 250
Branch Office Computer.
You never knew power
could be this simple.**



- Eloquence was first released in 1989 as a migration solution to move HP250/HP260 applications to HP-UX
- Since then Eloquence has gone through continuous development

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 9

Eloquence was first released in 1989 as a migration solution to move HP250/HP260 applications to HP-UX.

Since then Eloquence has gone through continuous development.

If you are with HP for a some time you might recognize this image. It comes from the cover sheet of an old HP250 sales brochure.

The HP250/HP260 system was targeted as an entry level business machine and discontinued in 1990.

A short history is available on the HP Eloquence web site.

<http://www.hp-eloquence.com/history/history.html>

Product Overview

The new Eloquence release is A.07.00

Major database enhancements include

- Substantially improved TurboIMAGE compatibility
- Native support of TurboIMAGE item types
- Forward recovery
- Increased Image limits
- Structural maintenance

The new Eloquence A.07.00 release was initially intended to add improvements such as

- A new Java and Web based user interface option for Eloquence applications
- Object support for the Eloquence programming language, including XML support
- Database enhancements, such as forward recovery and structural maintenance

However after the e3000 EOL announcement priorities were changed.

The new Eloquence release A.07.00 substantially improves the TurboIMAGE compatibility.

- Native support of TurboIMAGE item types
- Increased Image limits
- Support for the TPI and IMAGE b-tree modes

At the time of this writing (July 2002) the current release is A.06.31. Eloquence A.07.00 is currently available as a pre-release.

Ordering

The Eloquence product can be ordered

- from an Eloquence partner
- from the Eloquence web site or Marxmeier Software AG

The Eloquence product can be ordered from an Eloquence partner, from the Eloquence web site or Marxmeier Software. It can currently not be ordered directly from HP.

Since Eloquence is typically used as embedded part of a solution, the majority of Eloquence sales goes through partners that also provide added value to their customers, such as installation or support.

Product Options

- An Eloquence license is required for each server system
- The “unlimited license” option (AH0) provides an unlimited user license for a single system
- The “entry license” option (000) and additional user licenses allow competitive pricing for entry level configurations
- The corresponding TurboIMAGE compatibility option (3k) must be ordered in addition

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page 12

An Eloquence license is required for each server system. Two licensing models are available:

- The product option AH0 provides an unlimited user license for a single system.
- The product option 000 provides a two user entry license. Additional user licenses can be added (at any time) as required. This allows competitive pricing for entry level configurations. User licenses are added. For example, for a 26 user license, the options 000, 016 and 008 would be ordered.

In order to use the Eloquence TurboIMAGE compatibility extension the corresponding option 3k is required for each user license.

Detailed price and order information is available on the Eloquence web site.

Obtaining Eloquence

The Eloquence product can be obtained by download from the Eloquence web site

CD-ROM media can be ordered as an alternative

The Eloquence product is available for download from the Eloquence web site. This is the preferred distribution method. A separate product archive is available for each platform which is only a few megabytes in size.

A CD-ROM media can be ordered as an alternative

Product Evaluation

The “Personal Edition” provides a free two user license with a database up to 50 MB

A temporary license provides an unlimited version of Eloquence for evaluation purposes

A temporary license key can be requested from the Eloquence web site
<http://www.hp-eloquence.com/license>

There are three options to evaluate the Eloquence product

- The Eloquence “Personal Edition” provides a free and non-expiring two user license and is limited to a database size of 50 MB. The Eloquence “Personal Edition” license is intended for product evaluation and personal usage. This license key is installed by default.
- During the installation process, a temporary license key can be created (once per system) which is valid for a period of 30 days and which does not restrict the number of users or database size (option not available on the Linux platform).
- A temporary license key can be requested from the Eloquence web site at <http://www.hp-eloquence.com/license>

More Information

Detailed information is available on the Eloquence web site
<http://www.hp-eloquence.com>

TurboIMAGE migration
<http://www.hp-eloquence.com/hp3k>

Get in contact:
info@hp-eloquence.com

Detailed information on Eloquence, including ordering and pricing information, on-line documentation, patches and support related documents is available on the Eloquence web site at <http://www.hp-eloquence.com>

A separate section on the Eloquence web site is dedicated to TurboIMAGE compatibility and hp e3000 transition. It provides additional documentation including a “getting started” guide and the image3k mailing list.

To request additional information, please contact the Eloquence team at info@hp-eloquence.com

end of module 1
product overview

- Product overview
- Database concepts
- TurboIMAGE compatibility
- Database utilities
- Installation and configuration
- Data migration

This module provided a short overview on the Eloquence product.
Now we turn to the technical parts of the training ...

module 2 database concepts

- Product overview
- **Database concepts**
- TurboIMAGE compatibility
- Database utilities
- Installation and configuration
- Data migration

This module provides a short introduction to the Eloquence database concepts and explains some of the architectural differences to TurboIMAGE.

While the Eloquence database is compatible with TurboIMAGE at the application level, the underlying architecture is different.

The slide is divided into two main sections. On the left is an orange rounded rectangle containing the text 'database concepts'. On the right is a dark blue rounded rectangle containing a bulleted list of database concepts. At the bottom of the slide, there are four small text elements: '© 2002 Marxmeier Software AG', 'EQ HP3K Training.ppt', 'August 9, 2002', and 'page 18'.

database concepts

- Client/server architecture
- Indexes
- Locking
- Transactions
- Database names
- Database security
- Database environment

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 18

Important differences to TurboIMAGE are:

- The Eloquence database uses a client/server based architecture.
- Eloquence does not use hashing and comes with integrated indexing capabilities.
- Eloquence provides additional locking options, such as read level locking and selective unlock.
- Transactions can be nested, provide transaction isolation and are not limited in size.
- The Eloquence database uses an extended data base name which allows to specify the system on which the database is managed.
- The Eloquence database comes with a new, more flexible security subsystem
- The Eloquence database does not reside in the file system but is managed by a server process. The data and structural information is kept in volume files.
- Eloquence does not use a ROOT file. Structural information is kept in the server catalog inside the database.

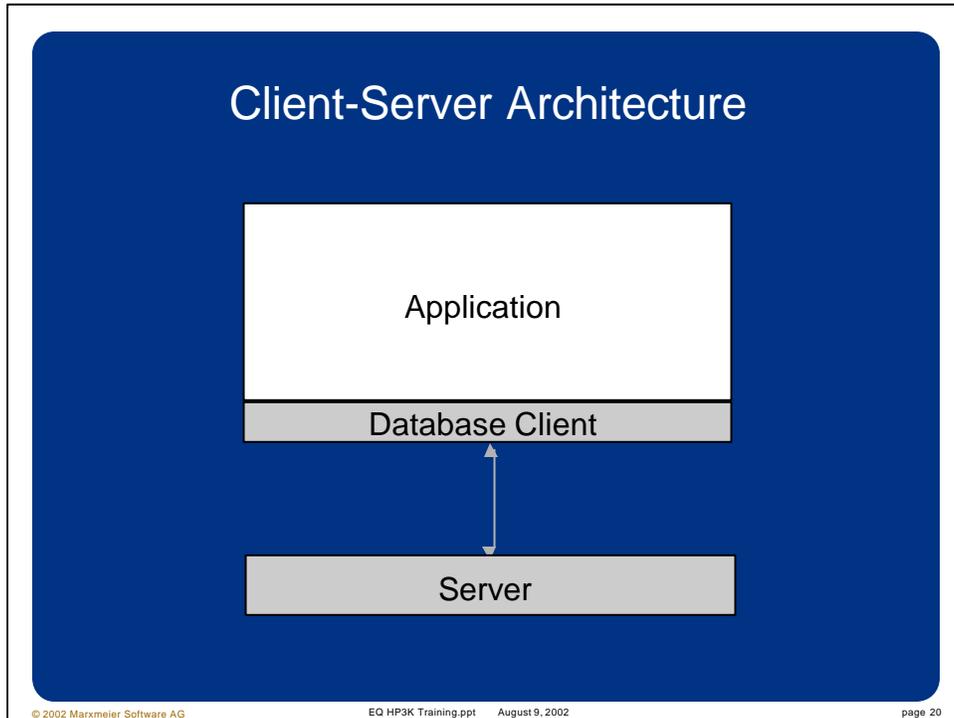
Client-Server Architecture

The Eloquence database uses a client/server based 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)

The Eloquence database uses a client/server based architecture. All database access is performed by the database server on behalf of the application.

The application is linked with the client library which provides the database API (Application Programming Interface). The client library submits any request to the database server which can not be processed locally and returns the results and any status codes to the application.



The figure above shows an application linked to the database programming interface which is connected to the database server.

The connection between the client side (application) and the server is made through the network. In case the application is running on the same system as the server, shared memory can be used to reduce the communication overhead.

When a database is opened, the server uploads structural information about the database to the client, so the client side is able to verify the call arguments and convert them as necessary. It also allows the client side to process most DBINFO calls locally.

Multiple Platforms

Eloquence is available for multiple operating systems and architectures

- HP-UX on PA-RISC
- Linux on Intel IA32
- Windows NT/2000/XP on Intel IA32

IA64 architecture will be added in 2003

The Eloquence database is available for different operating systems and hardware architectures.

The current version supports HP-UX on PA-RISC and Linux and Windows NT/2000/XP on the Intel IA32 architecture.

Future releases will add support for the IA64 (or IPF) architecture for the HP-UX, Linux and Windows operating systems.

Network Transparent

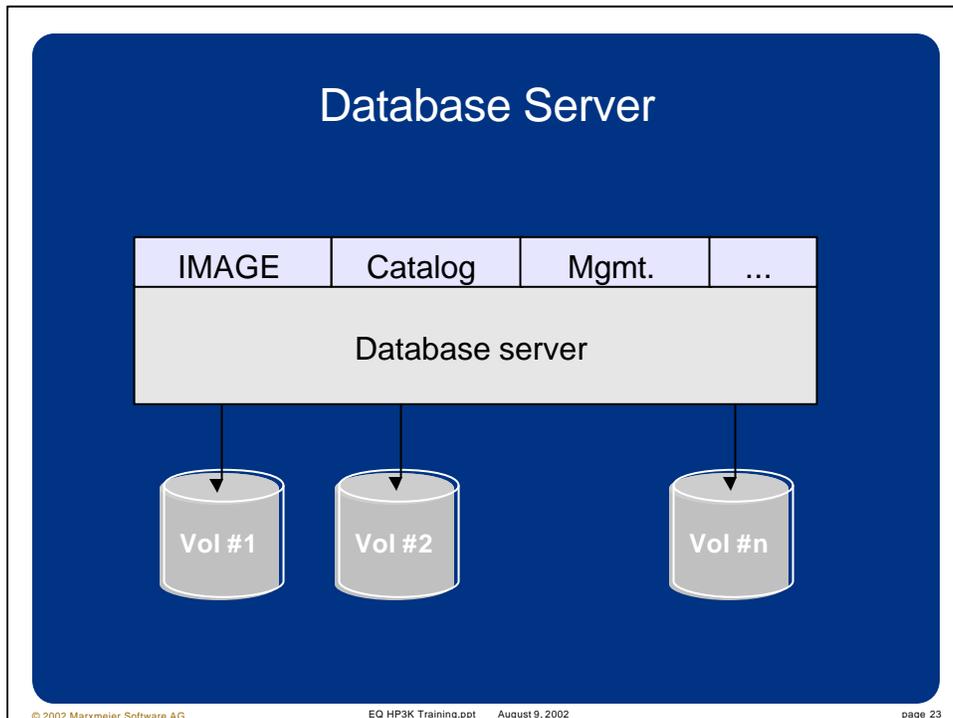
Applications running on different machines and operating systems can access a database

Requests and results are translated transparently

- Character set encoding
- Byte order conversion

The Eloquence database is network transparent and can be used in a heterogeneous environment. Applications running on different machines and operating systems can access a common database. The database server and client API make sure that requests and results are translated as necessary to match the specific platform requirements.

For example: PA-RISC systems uses a different byte order than Intel IA32 based systems. The Eloquence database client will translate any values transparently as required.



All database functionality is implemented by the database server. The client side implements the API (application programming interface) and submits the request to the server. The server processes the request and returns the results to the client.

The Eloquence database server uses a multi-threaded architecture. When a client connection is established, an internal thread is associated with the connection to process any client requests.

The server functions can be separated into groups. The most important ones are:

- The IMAGE API provides IMAGE procedures, such as DBPUT or DBGET. The database server implements IMAGE at the core.
- The Catalog API provides access to the server catalog. It is used by database utilities, for example the schema processor and dbdumpcat.
- The Management API provides access to server management functions. It is used by the dbctl utility.

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 is implemented in Eloquence 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

Compatibility goes beyond intrinsic calls and also includes performance profile

- Applications are built on assumptions and take advantage of specific behaviour
- An IMAGE application often expects that
 - a DBFIND or DBGET execute fast
 - DBGET performance does not differ substantially between modes
- Application may depend on external utilities or third party tools

Compatibility goes beyond intrinsic calls (and also includes a performance profile.)

Applications are built on assumptions and take advantage of specific behaviour. If those assumptions are no longer true, the application may still work but no longer be useful.

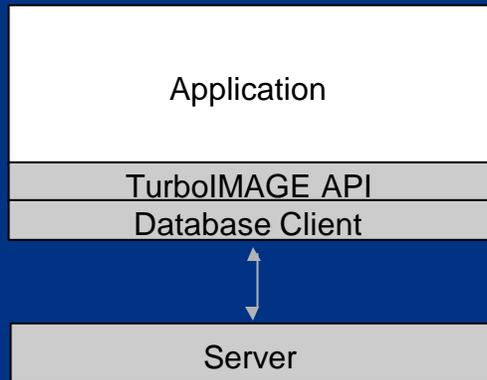
For example, an IMAGE application can reasonably expect that a DBFIND or DBGET execute fast, independently of the chain length and that DBGET performance does not differ substantially between modes.

If this is no longer true, the application may need to be rewritten, even though all intrinsic calls are available.

Applications may also depend on external utilities or third party tools. If your application relies on a specific tool (let's say, Robelle's SUPRTOOL) which is not available, significant changes to the application may be required.

Eloquence goes a long way to make sure that not only the intrinsic calls are compatible, but also reasonable expectations are met. We are working with hp e3000 solution providers to make sure that your familiar environment is available. Beyond the intrinsics.

TurboIMAGE compatibility



© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page 26

The TurboIMAGE compatibility API is implemented as a library on top of the native Eloquence database API. It does not impose a performance impact.

The TurboIMAGE compatibility API provides source code compatibility with existing applications and translates TurboIMAGE intrinsic calls to the Eloquence API. This involves different alignment requirements, modes and status codes, emulating different behavior or converting arguments as required.

Indexing

Eloquence comes with integrated indexing

- Uses indexes instead of hashing with master sets
- Supports both the TPI programming interface as well as IMAGE indexes ("super chains")

The Eloquence database comes with integrated indexing capabilities and Eloquence uses indexes rather than hashing with master sets.

Indexes are available to applications to implement indexed database access, such as partial key retrieval or ordered retrieval.

The TurboIMAGE compatibility extension allows an application to make use of indexes in a TurboIMAGE compatible manner.

With Eloquence A.07.00 both the TPI programming interface as well as TurboIMAGE indexes (used with master sets, "super-chains") are supported.

Indexing / TPI

Eloquence provides a commonly used subset of the TPI functionality

Advanced indexing functions are currently not available

- keywords
- relational access
- independent indexing

TPI (Third Party Indexing) is a specification of a common application call interface to use indexing products with TurboIMAGE.

Eloquence indexes can be used with the TPI modes. The TurboIMAGE compatibility API provides a commonly used subset of indexing functionality in a compatible way.

Advanced indexing functions are currently not available:

- keywords
- relational access
- independent indexing

Indexing vs. Hashing

Hashing is more efficient but less flexible

- Migrating secondaries require specific locking on master sets
- Applications must be aware of migrating secondaries
- Growing a master set causes hashing to become less efficient

The Eloquence database does not use hashing. While hashing is more efficient than indexes it is less flexible.

- Migrating secondaries (which are caused by hashing collisions) impose specific locking requirements for adding or deleting entries with master sets.
- Migrating secondaries must be accounted for in application programs when deleting entries from a master set or using record numbers with master sets.
- When extending the capacity of a master set either the hashing becomes less efficient or the data set must be reorganized (using a utility program).

Indexing vs. Hashing

Caching is used to improve index performance

Additional functionality is provided at no overhead

- partial key retrieval
- ordered retrieval
- wildcard lookup

Hashing is effective, because it often allows to access an entry with a single disk access. With indexes, first the key needs to be located in the index before the record can be retrieved.

Eloquence makes extensive use of caching to reduce the number of disk accesses. Since master sets often have a limited number of entries the caching is quite effective and can narrow the performance difference.

By using indexes instead of hashing Eloquence can also provide additional functionality at no overhead, such as partial key retrieval or ordered retrieval.

When an index is used with TurboIMAGE the performance advantage of using hashing is reversed, because the index needs to be maintained in addition.

Dynamic Dataset Expansion

- Capacity is no longer used
- Data sets grow dynamically when required
- Capacity value returned from DBINFO is highest record number ever used

With Eloquence, the data set capacity is no longer used. A data set grows dynamically when required. Since hashing is not used, performance is not affected, even for master sets.

When growing a data set, Eloquence allocates storage for a group of records (typically 64 Kb) and marks the extra entries as deleted.

The capacity value returned with DBINFO modes 202 and 205 indicates the highest record number ever used for this data set.

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 with the Eloquence database is fully compatible with TurboIMAGE but provides additional options.

The most visible difference is that locking is optional and the application is free to implement the locking strategy which suits best to its requirements. The Eloquence database neither depends on explicit locking (DBLOCK) nor does it impose restrictions on locking (like requiring a set lock for specific operations).

Instead of verifying if a write operation (such as DBPUT) is covered by a DBLOCK, Eloquence verifies if a conflicting DBLOCK has been granted which covers the same data. In this case a database status is returned.

This behavior is fully compatible since in TurboIMAGE all write operations must be covered by a DBLOCK and consequently there can be no concurrent DBLOCK.

Locking

Read locking and selective unlocking are available

- DBLOCK modes 11 to 16 implement read locking at the database, data set or item level
- DBUNLOCK allows to release a specific lock

Eloquence provides read locks and allows for selective unlocking.

DBLOCK modes 11 to 16 add read locking at the database, data set or item level, equivalent to the DBLOCK modes 1 to 6.

DBUNLOCK has additional modes which allow to selectively release a previously acquired lock by passing the same lock qualifier as used with DBLOCK.

Locking

- Multiple DBLOCKS are allowed
- If configured, locks on conflicting items are accepted
- Deadlock conditions are detected and a status is returned

Eloquence allows applications to use multiple DBLOCKS and is able to detect and act upon a deadlock.

A single DBLOCK can never cause a deadlock, but in case an application program has a DBLOCK and blocks while waiting for another one to be granted (unconditional DBLOCK) a deadlock condition could arise. For example, application 1 has a lock on the data set CUSTOMERS and waits to get a lock on PARTS while application 2 has a lock on PARTS and waits to get a lock on CUSTOMERS. This is detected by Eloquence and a status is returned.

Transactions can cause deadlock situations, if there are dependencies between concurrent transactions and locking is not used. Internally, Eloquence makes sure, that transactions can not overlap. In this case a database status is returned to the application which has the least work done in the current transaction.

To maintain full backwards compatibility, a deadlock condition is never returned to applications which do not make use of dynamic transactions or multiple locks.

When a deadlock status is returned, there is no use in retrying a statement. The appropriate action is to do a DBUNLOCK or a transaction rollback or to resolve the situation. Eloquence does not automatically free any resources in case a deadlock condition is detected.

Transactions

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

The Eloquence database relies on transactions to ensure data consistency.

Transactions in Eloquence are not specific to a database. All databases modified in a transaction are part of the transaction.

Eloquence does not expose incomplete transactions to concurrent processes. Changes in a transaction are only visible to the initiating process until the transaction is committed. This is called transaction isolation. If a concurrent process accesses database content modified in an incomplete transaction the original content is returned.

Incomplete transactions which overflow the cache are stored in the log volume and are only limited by the disk space available in the log volume.

All database procedures (such as DBPUT) are internally considered a transaction (implicit transactions) which are committed automatically once the call is completed. For example, a DBPUT can consist of 50+ internal operations and may need to be aborted cleanly at any time. We use the transaction mechanism for this.

If an application makes use of transactions (explicit transactions) any internal transactions become subtransactions and are merged on commit with the upper transaction level. Transactions can also be nested in applications and then behave the same way.

Transactions (cont)

Committed transactions are persistent

A checkpoint operation truncates the transaction journal in a regular interval

A committed transaction is guaranteed to be persistent, even if the server process terminates unexpectedly or the system crashes, for example due to a power failure.

Committed transactions are recorded in the log volume and are re-applied automatically when the server process is restarted. To protect against data loss in case of a system crash Eloquence has the option (if configured) to force the operating system to immediately write transaction records to disk.

The checkpoint operation is part of the transaction management. It is implemented as a thread which is scheduled regularly (each minute by default) or when the transaction journal in the log volume grows beyond a configurable size. The checkpoint operation truncates the transaction journal.

A committed transaction is initially noted in the log volume (to recover from a server abort) and actually written to the data volume(s) at a later time. The checkpoint operation makes sure all modified data pages (in the cache) are written back to the data volume(s) before the disk space used for the transaction journal is released.

Database Names

- A database name is no longer 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 instead of a file location

Eloquence uses an extended format for database names.

- The database name is no longer restricted to 6 characters and can include dots and hyphen characters in addition to letters and digits
- Eloquence databases do not reside (logically) in the file system but are managed by a server process. A database name does not specify a file location
- The database name can specify the network address of the database server as an option. To access a database on a remote system, simply add the server address to the database name

Database Names

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

- Hostname specifies database server system
- Service specifies database server instance

For example:

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

All specify the same database.

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page 38

The database name consists of the following elements:

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

- The host name or IP address which qualifies the system on which the database server resides. The default is the local system (localhost or IP address 127.0.0.1).
- The service name or port number which is used by the server process on that machine. The default is defined by the tcp service named eloqdb (which defaults to port 8102 during installation).
- The database name. If the host name or service is specified it is separated with a slash from the database name.

The example specifies the database SAMPLEDB on the local system, using the default database server instance in different forms:

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

Database Names

The EQ_DBSERVER environment variable can be used to specify a default server

Syntax

[hostname][:service]

- Hostname specifies database server system
- Service specifies database server instance

Example

EQ_DBSERVER=server:eloqdb2

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page 39

The environment variable EQ_DBSERVER can be used to specify the default database server. It has two parts:

[host][:service]

- The host name or IP address on which the database server resides. The default is the local system (localhost or IP address 127.0.0.1).
- The service name or port number which is used by the server process on that machine. The default is defined by the service name eloqdb (which defaults to port 8102 during installation).

For example:

If the EQ_DBSERVER environment variable is set to server:eloqdb2 then a DBOPEN will default to connect the database server running on the host named “server” at the port number defined by the tcp service “eloqdb2”.

Database Security

The database server maintains a list of users

Security groups specific to a database

- Similar to TurboIMAGE user classes
- Database privileges are assigned to groups
- A user can be a member of multiple groups

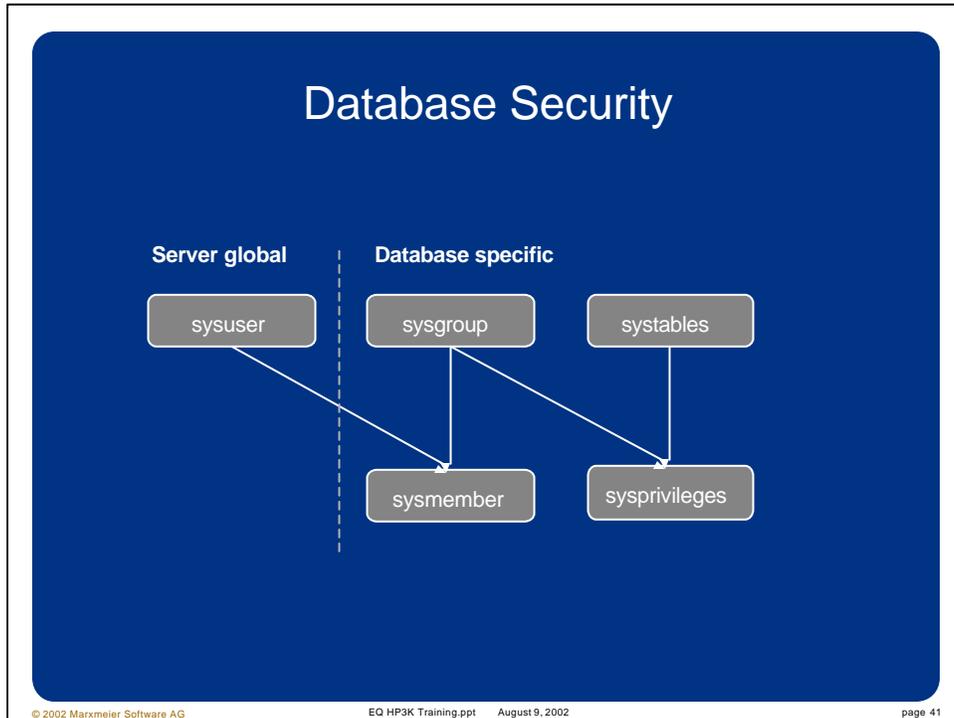
The Eloquence database server maintains its own user list. In order to connect to the Eloquence database server a user name and password is required. Two predefined users, "dba" and "public" are created by default.

- dba provides administrative access to the server and all databases but does not allow access to the data.
- public allows read/write access to the data but is not authorized to perform administrative tasks

Database access privileges are associated with a group. Groups are specific to a database. This is the equivalent of the TurboIMAGE user classes which are defined in the PASSWORDS: section of the schema file.

The schema processor creates two default groups "public" and "dba". The group "dba" provides the privilege to erase a data base, the group "public" is the equivalent to the user class 0 in TurboIMAGE. If specified in the schema file, additional groups are created with the privileges defined in the schema file.

A user can be a member of up to eight groups and gets the added privileges associated with the groups. Unless associated with a group, no access to the database is possible. By default, the schema processor associates the user "dba" with the group "dba" and the user "public" with the group "public" and all other groups defined in the schema file. This can be changed with the dbutil utility.



This figure provides an overview how the different elements are related.

- At the server scope, the database server maintains a list of users
- For each database privilege groups are maintained
- Access privileges are defined per data set and are associated with groups
- Users can be a member of multiple groups

When a database is opened, the groups the user is a member of are determined and the combined privileges of the groups are associated with the user to define the database access privileges.

Database Security

A file can be specified which contains the user and password

The EQ_DBUSER environment variable can be used to specify a default user

The EQ_DBPASSWORD environment variable can be used to specify a default password

Example

```
EQ_DBUSER="mike"  
EQ_DBPASSWORD="file:/home/mike/db/pswd"
```

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page 42

The user name or password can be a reference to a file which holds the actual user name and/or password.

- If the user name starts with file: it is expected to be a reference to a file. The first line in this file is used as the user name and if present a second line is used as the password.
- If the password starts with file: it is expected to be a reference to a file. The first line in that file is used as the user password.

Using a file makes sure the password is not exposed to other users which can use the ps utility to obtain the command line of executing processes or may have access to environment variables (on Linux through the ps utility or the /proc file system).

The environment variables EQ_DBUSER and EQ_DBPASSWORD can be used to specify the default user and password.

For example:

If the environment variable EQ_DBUSER is set to "mike" and the EQ_DBPASSWORD variable is set to "file:/home/mike/db/pswd" and the file /home/mike/db/pswd holds the string "secret" the user name "mike" and the password "secret" are used by default to access the database.

Database Security

The new DBLOGON procedure can be used to specify user and password

With the TurboIMAGE compatibility API the DBOPEN password argument can be used to specify a user and password

The EQ_DBUSER and EQ_DBPASSWORD environment variables can specify a default user and password

Eloquence has the new DBLOGON call to provide authorization information when connecting to a database server.

By default Eloquence uses the user "public" which is the equivalent to the UCL 0 with TurboIMAGE. For administrative tasks (such as schema) Eloquence utilities default to the user "dba".

Authorization information can be provided in different ways:

- You can explicitly use the (Eloquence specific) DBLOGON call to specify a user name and password which is then used with a subsequent DBOPEN.
- With the TurboIMAGE compatibility API you can use the DBOPEN password argument to specify a user and password. This will result in an implicit call to DBLOGON.
- You can use the EQ_DBUSER and EQ_DBPASSWORD environment variables to setup a different default

Once a user and password has been specified it is saved and used with subsequent DBOPEN.

The slide features a large orange rounded rectangle on the left containing the text 'database environment'. To its right is a dark blue rounded rectangle containing a bulleted list of five points. At the bottom of the slide, there is a thin line with small text indicating copyright and page information.

database environment

- A database environment is an organizational entity
- It can manage any number of databases
- Database names must be unique
- Databases share common resources
- Multiple database environments can coexist on the same system

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 44

A database environment is an organizational entity or domain.

It represents a database server instance which can manage any number of databases.

Within an environment database names must be unique.

The databases in an environment share common resources.

Multiple database environments can coexist on the same system.

To put it differently, a database environment is equivalent to a database server and everything which is required to operate it.

Database Environment

A database environment consists of

- a configuration file
- a primary data volume
- a transaction log volume

Multiple database environments can coexist on the same machine, each managed by a separate server process

Database Environment

The configuration file defines

- the server configuration
- scaling and tuning parameters
- the location of the volume files

The server configuration file is a text file, similar to INI file format, which specifies the server configuration, scaling and tuning parameters and the location of the volume files. It is read when the server is started.

A leading hash sign indicates a comment. The file has multiple sections with the section names enclosed in square brackets.

Volumes

- 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)
- Volume files are specific to the system architecture (byte order)

The Eloquence database does not reside in the file system but uses volume files which reside in the file system as a container for the data and structural information.

Volume files are a storage container managed by the database server. All volumes combined define the disk space accessible to the database server.

The maximum size of a single volume file is 128 GB. As of Eloquence A.07.00 a single volume file is limited to 2 GB on the HP-UX and Linux platform (we currently don't use 64 bit file operations on HP-UX and Linux to maintain compatibility with older operating system versions). On the Windows platform we support 64 bit file operations so the 2 GB limit for a single volume does not apply.

A maximum of 255 volume files is supported by a single server instance which defines an upper limit of about 500 GB for the current version and up to 32 TB once the 2 GB limit has been removed.

Volume files are specific to the system architecture (byte order).

Volumes

A volume file is internally maintained in 8 KB blocks

Consecutive blocks can be used to form a cluster

The first block in a volume contains additional volume information

All volume files in an environment share the same serial number and generation

The order in the server configuration file is not significant

Internally, volume files are organized as a collection of 8 KB blocks (or clusters built from them). The first block in a volume is used to hold information about the volume and provide anchors to volume specific information (like the block allocation map).

Volume files for a database environment share a common serial number and generation and contain ordering information so the order in which they are specified in the eloqdb6.cfg configuration file is not significant.

Volumes

Different volume types

- Primary data volume
- Transaction log volume
- Additional data volume(s)

Three different volume types exist:

- Primary data volume
- Transaction log volume(s)
- Additional data volume(s)

Volumes

The primary data volume contains the server catalog and the database contents

Additional data volumes can be added to extend the primary data volume

The primary data volume holds the initial server catalog and is created with the dbvolcreate utility.

Additional data volume(s) are used to add additional disk space to make additional disk space available. A secondary data volume is added to the database environment with the dbvoextend utility.

Volumes

A log volume contains

- a journal of recently committed transactions
- information on incomplete transactions
- temporary disk space during online backup

The transaction journal is used in case of an unexpected server abort to perform a recovery of recent transactions

The transaction log is truncated when it reaches a configured size

The transaction log volume holds recovery information which is used in case of an unexpected termination of the server process and provides temporary storage for uncommitted transactions and during on-line backup (in theory, more than one log volume could be used but this should hardly be necessary). A log volume is added to the database environment with the `dbvolestend` utility.

The checkpoint operation is part of the transaction management. It truncates the transaction journal and is scheduled at a regular interval (each minute by default) or when the transaction journal (in the log volume) grows beyond a configurable size.

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

Eloquence does not use a ROOT file. Database structural information is maintained in the database environment, called the server catalog. The catalog is an internal database maintained by the server process.

The server catalog is created and initialized with the dbvolcreate utility (which is used to create the primary data volume) and maintained with database utilities, such as the schema processor or the dbutil utility. The dbdumpcat utility allows to read the server catalog.

Server Catalog

The server catalog contains global and database specific information

Global catalog information

- list of users and databases

Database specific information

- schema and security information

There are two scopes of catalog tables. Global tables apply to the database environment (server instance) while database specific views are specific to a single database.

Global catalog tables include:

sysuser	database users
sysdb	list of databases managed by this server

Database specific catalog views include:

systables	data sets in a specific database
sysgroup	security groups for a database
syscolumns	data items for a specific database
sysindex	indexes defined for a database.
sysmember	users associated with a security group
sysprivilege	privileges granted to a security group

Online Backup

In online backup mode, the data volumes are frozen

Modifications during online backup are temporarily saved into the transaction log volume

Any backup software can be used to create a consistent backup

The Eloquence database supports online backup.

When entering online backup mode, the database server makes sure the data volume(s) are consistent and no longer changed, even if the database is in use. All database modifications are saved temporarily in the log volume.

This makes sure any backup software can be used to create a consistent backup without interrupting ongoing work and allows for easy and straight forward integration into backup procedures. Since the entire database environment is backed up, the backup will cover all databases which are managed in that environment.

When finishing the online backup mode pending changes are copied to the data volume(s).

For example:

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

Online Backup

The dbctl utility is used to control online 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
```

The database is put in online backup mode by sending the backup start request with the dbctl utility. In this example the tar utility is used to create a backup of the database environment. After tar has completed, the backup mode is finished by sending the backup stop command.

During the backup, the data volumes are frozen and do not change. The transaction log volume does change since it is used for temporary storage, but the content is not used and saving it in the backup is optional. It can be recreated after a restore.

Forward Logging

Forward logging is used to record all modifications since a previous backup

Forward recovery is fast and involves only minimal processing

The forward log files can be managed automatically by the server process

The database server provides the option to log all changes since a previous (online or offline) backup to a file, device or process.

This log can be applied with the `dbrecover` utility to recover all modifications since a previous backup. The forward recovery is fast and involves only minimal processing.

As an option, the database server can manage forward log files automatically. During backup, a new log file is created. Removal of old log files can easily be integrated into the backup procedure.

Database Limits

Eloquence A.07.00 Image limits

- 2048 data items
- 500 data sets
- 64 / 16 paths
- Entry length 5120 bytes

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page 57

The Eloquence A.07.00 database either matches or exceeds the TurboIMAGE schema limits.

Most TurboIMAGE limits do not apply to Eloquence because the underlying architecture is different and Eloquence allocates most resources dynamically.

Although the Eloquence database may exceed the TurboIMAGE limits in some cases, you should be careful to make use of it. Applications may need to be reviewed, if they support the increased limits. As the IMAGE intrinsics do not specify buffer sizes (this is especially important with DBINFO as it returns a variable sized result) this may result in application crashes due to a buffer overflow. With the Eloquence TurboIMAGE compatibility API DBINFO mode 406 returns information about database schema size.

Disk space within the volume files is allocated dynamically. A single data set can span multiple volumes and is therefore not limited to the size of a single volume file. Transactions can grow dynamically and are only limited by available disk space in the log volume(s).

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

- Recommended max. number of concurrent users is 500 for the A.07.00 release

Some Eloquence limits apply to a database environment rather than a specific database as all databases managed by a database environment share common resources.

The size of a database or data set is only limited by the disk space available in the volume files

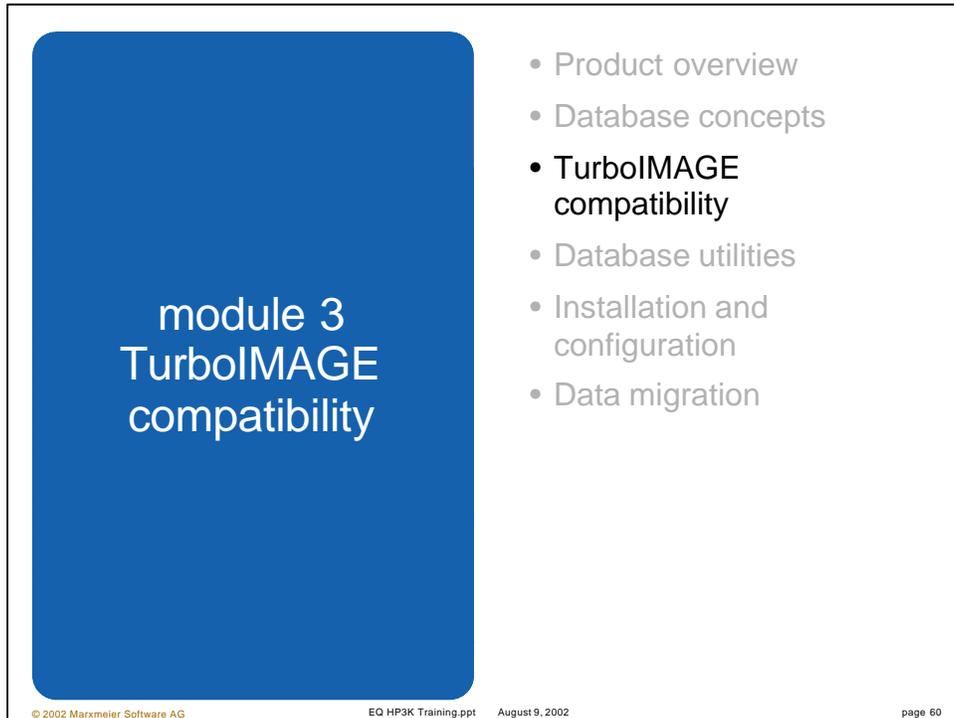
- The maximum disk space which can be allocated to a database environment is currently limited to about 500 GB. This is imposed by the current limitation of 2 GB per volume file.
- The maximum amount of disk space which can be managed by a database environment is about 32 TB, which should be sufficient for a while.

The number of concurrent users per database environment is limited to 1000. However we do not recommend to use more than 500 concurrent users per database environment with the A.07.00 release.

end of module 2 database concepts

- Product overview
- **Database concepts**
- TurboIMAGE compatibility
- Database utilities
- Installation and configuration
- Data migration

This module provided a short introduction to the Eloquence database architecture and covered some of the conceptual differences to TurboIMAGE.



The slide features a blue rounded rectangle on the left containing the text "module 3 TurboIMAGE compatibility". To the right is a bulleted list of topics. At the bottom, there is a footer with copyright information and page number.

- Product overview
- Database concepts
- **TurboIMAGE compatibility**
- Database utilities
- Installation and configuration
- Data migration

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 60

In this module we will have a detailed look at the Eloquence TurboIMAGE compatibility.

We compare the TurboIMAGE intrinsics with the Eloquence native modes and have a look at the functionality supported by the Eloquence TurboIMAGE compatibility API.

We also have a look at differences and limitations of the Eloquence TurboIMAGE compatibility which are caused by architectural differences.

The slide features a large orange rounded rectangle on the left containing the text "TurboIMAGE compatibility". To its right is a dark blue rounded rectangle containing three bullet points. At the bottom of the slide, there is a thin line with small text: "© 2002 Marxmeier Software AG", "EQ HP3K Training.ppt", "August 9, 2002", and "page 61".

TurboIMAGE compatibility

- The TurboIMAGE compatibility adds the TurboIMAGE intrinsics to the Eloquence database
- All TurboIMAGE intrinsics and almost all modes are supported and behave identical
- HP e3000 applications can usually be ported with only minor changes

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 61

The Eloquence TurboIMAGE compatibility adds a TurboIMAGE compatible programming interface to the Eloquence database.

The Eloquence database implements IMAGE operations at the database core. However the Eloquence application programming interface differs in some details.

The TurboIMAGE compatibility API translates the TurboIMAGE intrinsics to Eloquence and makes sure TurboIMAGE behavior is matched

All TurboIMAGE intrinsics and almost all modes are supported and behave identical.

HP e3000 applications can usually be ported with only minor changes

Currently not supported

- Some DBCONTROL modes
 - DBCONTROL modes which are specific to TurboIMAGE implementation details
 - DBCONTROL modes for b-tree maintenance
- Item level security
 - All items are considered writable
- Static transactions
 - Eloquence currently does not support static transactions
 - The DBBEGIN, DBEND and DBMEMO intrinsics have no effect

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page 62

While the Eloquence TurboIMAGE compatibility provides a high degree of compatibility to TurboIMAGE some functionality is currently not supported, because the Eloquence architecture has no equivalent:

- Only a limited number of DBCONTROL modes are supported. The CIUPDATE and btreemode1 can be set. Other DBCONTROL modes either have no effect or return a status code.
- Item level security is not supported, all items are considered writable
- Eloquence relies on transactions for database consistency. There is no equivalent to static transactions and the DBBEGIN, DBEND and DBMEMO intrinsics have no effect.

Transactions

The Eloquence transaction management is different from TurboIMAGE

- Transactions cover all databases
- Transaction isolation does not expose uncommitted changes to concurrent sessions
- Transactions can be nested

The Eloquence transaction management is different from TurboIMAGE. Eloquence behaves identical to TurboIMAGE if dynamic transactions are not active.

- Transactions in Eloquence are not database specific. All database modifications after beginning a transaction are part of the transaction until the the transaction ends.
- Eloquence does not expose incomplete transactions to concurrent processes. Changes in a transaction are only visible to the initiating process until the transaction is committed (transaction isolation).
- Eloquence transactions can be nested and do not have a fixed limit in size.

Locking

Eloquence provides additional locking options

- Database modifications need not be covered by a DBLOCK
- If a concurrent DBLOCK is active, database status is returned
- Additional DBLOCK and DBUNLOCK modes are available

Eloquence locking provides additional options

- Instead of verifying if a DBLOCK covers the current modification, Eloquence checks if a concurrent DBLOCK is active. This is fully compatible, since with TurboIMAGE all modifications must be covered by a DBLOCK.
- Eloquence supports additional DBLOCK and DBUNLOCK modes. DBLOCK supports READ level locking and DBUNLOCK selective unlocking.

TPI Compatibility

Eloquence implements basic TPI functionality

Currently not supported TPI functionality

- Keywords
- Relational access
- Independent indexing

TPI (Third Party Indexing) is a specification of a common application call interface to use indexing products with TurboIMAGE.

Eloquence indexes can be used with the TPI modes. The TurboIMAGE compatibility API provides a commonly used subset of indexing functionality in a compatible way.

Advanced indexing functions are currently not available:

- keywords
- relational access
- independent indexing

Schema processor

The Eloquence schema processor accepts HP e3000 schema files

The -T option specifies TurboIMAGE compatibility

The Eloquence schema processor is compatible with TurboIMAGE schema files. If the option -T is passed to schema, the schema file is assumed to follow TurboIMAGE conventions.

Most \$CONTROL options in the schema file have no effect and are ignored.

The Eloquence schema processor may fail to process a TurboIMAGE schema file, if the syntax does not follow the documented format although it was valid with the the TurboIMAGE schema processor.

Schema processor

Item types J and R are not supported natively and are only visible from the TurboIMAGE compatibility mode

- J is mapped to I
- R is mapped to E (IEEE format)

Item types J and R are not supported natively by Eloquence. They are mapped to equivalent item types.

- The J item type is mapped to the I item type.
- The R item type is mapped to the E item type, any values are expected in IEEE floating point format.

The J and R item types are accepted by the schema processor and are returned by DBINFO 102 when using the TurboIMAGE Compatibility API.

Database names

Databases do not reside in the file system but are managed through a server process

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

- Hostname specifies database server system
- Service specifies database instance

Eloquence can use an extended format for database names.

- The database name is no longer restricted to 6 characters. It has a variable length and can include dots and hyphen characters in addition to letters and digits
- Since Eloquence databases do not reside (logically) in the file system but are managed by a server process, a database name does not specify a file location
- The database name can specify the network address of the database server as an option. To access a database on a remote system, simply add the server address to the database name or define the EQ_DBSERVER environment variable

The default database server can be specified with the EQ_DBSERVER environment variable, otherwise the default server instance (localhost:eloqdb) is used.

Database security

- A user name and password are required to connect a database server
- A default user is used if no specific user is specified
- Different ways to specify user and password
 - The DBLOGON procedure can be used to specify user and password
 - The DBOPEN password argument can be used to specify the password and user
 - The EQ_DBUSER and EQ_DBPASSWORD environment variables can be used to specify the default user and password

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page 69

The Eloquence database server maintains its own user list. In order to connect to the Eloquence database server a user name and password is required. Two predefined users, "dba" and "public" are created by default.

Authorization information can be provided in different ways:

- You can explicitly use the DBLOGON call to specify a user name and password which is then used with a subsequent DBOPEN.
- With the TurboIMAGE compatibility API you can use the DBOPEN password argument to specify the password and user. If both are present, DBLOGON is called implicitly.
- The environment variables EQ_DBUSER and EQ_DBPASSWORD can be used to specify the default user and password.

The user name or password can be a reference to a file which holds the actual user name and/or password.

DBOPEN database name

<p><u>IMAGE</u></p> <ul style="list-style-type: none"> • Basename __ basename[.group[.account]] • Password password[/USERIDENT] 	<p><u>Eloquence</u></p> <ul style="list-style-type: none"> • Basename [[server][:service]/]basename • Password (ignored) <p><u>TurboIMAGE compatibility</u></p> <ul style="list-style-type: none"> • Basename __ [[server][:service]/]basename • Password password[/user]
---	---

© 2002 Marxmeier Software AG
EQ HP3K Training.ppt August 9, 2002 page 70

The first intrinsic an application needs to call is DBOPEN.

The first parameter of DBOPEN is the database name

- With TurboIMAGE the database name is limited to 6 characters and a group and account can be specified as an option. Two leading spaces need to be provided which are replaced with the database id.
- Eloquence supports a variable sized database name and the basename argument must be terminated.
- With the Eloquence TurboIMAGE compatibility two leading spaces must be specified which are replaced with the database id.

The password argument is the second argument.

- With TurboIMAGE the database password is specified.
- With Eloquence, the password argument is ignored and DBLOGON is used to provide authorization information.
- With the Eloquence TurboIMAGE compatibility, the password and user name can be specified by providing the optional user identifier. The password argument is ignored unless the optional user identifier is specified and a default login (“public”) is used.

DBOPEN modes

IMAGE

Mode 1: modify DB, allow concurrent 1,5
Mode 2: update DB, allow concurrent 2,6
Mode 3: modify DB, exclusively
Mode 4: modify DB, allow concurrent 6
Mode 5: read DB, allow concurrent 1,5
Mode 6: read DB, allow concurrent 2/4/6/8
Mode 7: read DB, exclusively
Mode 8: read DB, allow concurrent 6/8

Eloquence

Mode 1: modify DB, allow concurrent read/modify
Mode 3: modify DB, exclusively
Mode 8: read DB, allow concurrent read
Mode 9: read DB, allow concurrent read/modify

TurboIMAGE has 8 DBOpen modes which allow various sharing combinations.

Eloquence simplifies the DBOpen modes and provides 4 modes:

- With DBOpen mode 1, the database is opened read/write.
- Mode 3 opens the database exclusive.
- Mode 8 opens the database read-only and does not allow a concurrent write.
- Mode 9 opens the database read-only.

DBOPEN modes (compatibility)

- TurboIMAGE modes 1, 3 and 8 map to the same Eloquence modes and behave the same
- TurboIMAGE modes 2 and 4 map to Eloquence mode 1 which is similar but allows additional concurrent DBOPEN modes
- TurboIMAGE modes 5 and 6 map to Eloquence mode 9 which allows any non-exclusive concurrent DBOPEN modes
- TurboIMAGE modes 7 maps to Eloquence mode 8 which allows concurrent read-only DBOPEN modes

The Eloquence TurboIMAGE compatibility supports the TurboIMAGE DBOPEN modes.

Mode 1, 3 and 8 are mapped to same mode in Eloquence. Mode 2 and 4 are mapped to mode 1 of Eloquence. Mode 5 and 6 are mapped to mode 9 of Eloquence and mode 7 is mapped to mode 8 of Eloquence

Mode 1: modify DB, same as Eloquence mode 1
Mode 2: modify DB, map to Eloquence mode 1
Mode 3: modify DB, same as Eloquence mode 3
Mode 4: modify DB, map to Eloquence mode 1
Mode 5: read DB, map to Eloquence mode 9
Mode 6: read DB, map to Eloquence mode 9
Mode 7: read DB, map to Eloquence mode 8
Mode 8: read DB, same as Eloquence mode 8

DBCLOSE modes

IMAGE

Mode 1: close the database
Mode 2: close the dataset file but still holds the lock
Mode 3: rewind the dataset

Eloquence

Mode 1: close the database
Mode 3: rewind the dataset

TurboIMAGE compatibility

- Mode 1: same
- Mode 2: map to mode 3 of Eloquence
- Mode 3: same

TurboIMAGE has 3 DBCLOSE modes.

Eloquence supports DBCLOSE modes 1 and 3. DBCLOSE mode 2 does not apply to Eloquence.

The Eloquence TurboIMAGE compatibility supports all TurboIMAGE DBCLOSE modes. DBCLOSE mode 2 is mapped to the Eloquence DBCLOSE mode 3.

DBCLOSE will not abort an ongoing transaction with Eloquence. Transactions in Eloquence are not database specific.

DBFIND modes

IMAGE

Mode 1: non-B-tree find or B-tree find for X and U type

Mode 4: B-tree find for all item types

Mode 10: same as old DBFIND, does not use B-tree even if B-tree exist

Mode 21: same as mode 1 except the chain information is not accurate

Mode 24: same as mode 4 except the chain information is not accurate

Eloquence

Mode 1: find chain head/first match for search item or index item

Mode 2: find first match for index item

Mode 3: find last match for index item

Mode 4: find first match for index item with regular expression

Mode 5: find last match for index item with regular expression

Mode 6: find matching entries in index (list of conditions passed), status array returns no. of qualified records

Mode 7: find matching entries in index, status array will not return no. of qualified records

TurboIMAGE supports 5 DBFIND modes. DBFIND modes 1 and 10 are used to get the chain head of a search item. Modes 1,4,21 and 24 are related to TurboIMAGE b-tree find. Additional DBFIND modes are available if a TPI product is installed.

The Eloquence DBFIND mode 1 is used to locate a chain head when used with a search item or can be used with an index. The Eloquence DBFIND modes 2 to 7 are used with indexes.

DBFIND modes (compatibility)

IMAGE modes

Mode 1: non-B-tree find

Mode 10: same as old DBFIND, does not use B-tree even if B-tree exists

IMAGE B-tree modes ("super-chains")

Mode 1: B-tree find for X and U type

Mode 4: B-tree find for all item types

Mode 21: same as mode 1 except the chain information

Mode 24: same as mode 4 except the chain information

The Eloquence TurboIMAGE compatibility supports all TurboIMAGE DBFIND modes.

The TurboIMAGE b-tree DBFIND modes are supported. When DBFIND is used with a search item and btreemode1 is enabled, an index lookup is performed. Since Eloquence uses indexes instead of hashing with master sets, enabling indexed access does not cause a significant overhead.

DBFIND modes 4 and 24 can also be used with Eloquence indexes.

DBFIND TPI modes (compatibility)

TPI modes

Mode 1: TPI DBFIND

Mode 11: TPI range DBFIND for binary data items

Mode 21: same as DBFIND mode 1 except entry count

Mode 22: same as DBFIND mode 11 except entry count

Modes 1xx to 5xx, TPI DBFIND for =,>,>=,<,<= relation

The Eloquence TurboIMAGE compatibility supports a commonly used subset of the TPI functionality.

Some TPI DBFIND modes are supported to provide partial key retrieval, wildcards and ordered retrieval with Eloquence indexes.

When used with an index, DBFIND mode 1 follows the TPI conventions. DBFIND modes 4 and 24 can also be used with indexes in addition to the TPI modes.

DBGET modes

<p><u>IMAGE</u></p> <p>Mode 1: re-read, retrieves the current rec.</p> <p>Mode 2: serial read, forward</p> <p>Mode 3: serial read, backward</p> <p>Mode 4: direct read, using the record number in argument parameter</p> <p>Mode 5: chain read, forward</p> <p>Mode 6: chain read, backward</p> <p>Mode 7: calculated read, master only</p> <p>Mode 8: calculated read, primary location</p>	<p><u>Eloquence</u></p> <p>Mode 1: same as IMAGE mode 1</p> <p>Mode 2: same</p> <p>Mode 3: same</p> <p>Mode 4: same</p> <p>Mode 5: same, also for index access</p> <p>Mode 6: same, also for index access</p> <p>Mode 7: same</p> <p>Mode 15: next in index order</p> <p>Mode 16: previous for index order</p>
---	--

© 2002 Marxmeier Software AG
EQ HP3K Training.ppt
August 9, 2002
page 77

TurboIMAGE has 8 DBGET modes. Additional modes are available when a TPI product is installed.

The Eloquence DBGET modes 1 to 7 are the same as in TurboIMAGE. DBGET mode 8 does not apply since Eloquence does not use hashing.

The Eloquence DBGET modes 5,6,15 and 16 can be used with indexes.

DBGET modes (compatibility)

- TurboIMAGE modes 1 to 7 map to the same Eloquence modes and behave the same
- Eloquence does not have migrating secondaries so TurboIMAGE mode 8 is identical to mode 7
- Eloquence supports TPI modes 15 and 16

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page 78

The Eloquence TurboIMAGE compatibility supports all TurboIMAGE DBGET modes. Since Eloquence does not use hashing DBGET mode 8 is mapped to DBGET mode 7.

The DBGET modes 5, 6, 15 and 16 are supported and used to retrieve the next or previous record in index order after a DBFIND on an index.

Mode 1: re-read, same

Mode 2: serial read, same

Mode 3: serial read, same

Mode 4: direct read, same

Mode 5: chain read, same

Mode 6: chain read, same

Mode 7: calculated read, same

Mode 8: calculated read, same as mode 7

DBPUT modes

IMAGE
Mode 1: the only mode for DBPUT

Eloquence
Mode 1: the only mode for DBPUT

TurboIMAGE compatibility

- Mode 1: same

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 79

Both TurboIMAGE and Eloquence have only DBPUT mode 1.
The functionality is the same, however with Eloquence locking is optional and a status is returned if a concurrent DBLOCK is active.

DBDELETE modes

IMAGE

Mode 1: the only mode for DBDELETE

Eloquence

Mode 1: the only mode for DBDELETE

TurboIMAGE compatibility

- Mode 1: same

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 80

Both TurboIMAGE and Eloquence have only DBDELETE mode 1.
The functionality is the same, however with Eloquence locking is optional and a status is returned if a concurrent DBLOCK is active.

DBUPDATE modes

IMAGE
Mode 1: the only mode for DBUPDATE

Eloquence
Mode 1: for normal DBUPDATE
Mode 2: for critical item update

TurboIMAGE compatibility

- If CIUPDATE mode is active (DBCONTROL mode 5) mode 1 is mapped to mode 2

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 81

TurboIMAGE has only DBUPDATE mode 1. However DBUPDATE behavior depends on the CIUPDATE flag.

The Eloquence DBUPDATE has two modes. Mode 1 does not update search items and mode 2 is equivalent to the TurboIMAGE CIUPDATE mode.

The Eloquence TurboIMAGE compatibility supports both DBUPDATE mode 1 and 2.

If the CIUPDATE flag is set through DBCONTROL mode 5, the Eloquence TurboIMAGE compatibility will map DBUPDATE mode 1 call to Eloquence mode 2.

With Eloquence locking is optional and a status is returned if a concurrent DBLOCK is active.

DBINFO modes 1xx

<p><u>IMAGE</u></p> <p>Mode 101: type of access for an item</p> <p>Mode 102: describes an item</p> <p>Mode 103: items in database</p> <p>Mode 104: items in data set</p> <p>Mode 113: Btreemode1 & wildcard</p>	<p><u>Eloquence</u></p> <p>Mode 101: item number for a given item</p> <p>Mode 102: describes an item</p> <p>Mode 103: items in database</p> <p>Mode 104: item in data set</p> <p><u>TurboIMAGE compatibility</u></p> <ul style="list-style-type: none"> • Modes 101 to 104 and 113 are supported • Modes 101, 103 and 104 will always return negative item number when the database is opened in read-write mode
---	--

© 2002 Marxmeier Software AG
EQ HP3K Training.ppt
August 9, 2002
page 82

TurboIMAGE supports a number of DBINFO modes.

The Eloquence TurboIMAGE compatibility supports the TurboIMAGE DBINFO modes in addition to Eloquence specific and TPI specific DBINFO modes.

This slide shows the TurboIMAGE 100s DBINFO modes.

Eloquence supports the DBINFO modes 101 to 104.

The TurboIMAGE compatibility supports the DBINFO modes 101 to 104 and 113.

Since Eloquence does not support item level security, DBINFO modes 101, 103 and 104 will always return negative item numbers when the database is opened in read-write mode

DBINFO modes 2xx

<p><u>IMAGE</u></p> <p>Mode 201: type of access for data set</p> <p>Mode 202: describes a data set</p> <p>Mode 203: sets in database</p> <p>Mode 204: sets with item</p> <p>Mode 205: set capacity</p> <p>Mode 206: no. of data set chunks</p> <p>Mode 207: size of each chunk for a set</p> <p>Mode 208: primary/current/max cap</p> <p>Mode 209: B-tree attachment</p>	<p><u>Eloquence</u></p> <p>Mode 201: type of access for data set</p> <p>Mode 202: describes a data set</p> <p>Mode -202: describes a data set, same as 202 but processed locally</p> <p>Mode 203: sets in database</p> <p>Mode 204: sets with item</p> <p><u>TurboIMAGE compatibility</u></p> <ul style="list-style-type: none"> • Modes 201 to 209 are supported • Modes 202, 205: capacity is highest record number
--	---

© 2002 Marxmeier Software AG
EQ HP3K Training.ppt
August 9, 2002
page 83

This slide shows the TurboIMAGE 200s DBINFO modes.

Eloquence has the DBINFO modes 201 to 204. Mode -202 is used with Eloquence to return set information without the number of entries or capacity values (so it can be processed locally).

The TurboIMAGE compatibility supports the DBINFO modes 201 to 209.

Since Eloquence does use the data set capacity, the capacity value returned with DBINFO modes 202 and 205 is the highest record number ever used with this data set. This value reflects the disk space allocated for this data set.

DBINFO modes 2xx (compatibility)

Mode 201: same
Mode 202: blocking factor does not apply, always 0
Mode 203: same
Mode 204: same
Mode 205: some elements return static value
Mode 206: jumbo does not apply, always 0
Mode 207: jumbo does not apply, 0 for element 1 and 2
Mode 208: most elements return static value
Mode 209: same

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page 84

Since the Eloquence architecture differs from TurboIMAGE, some DBINFO modes return information which does not apply to Eloquence. In this case the Eloquence TurboIMAGE compatibility returns static values.

For mode 205:

element 11, block factor is 0;
element 18-19, high water mark is 0;
element 20-21, maximum capacity returns $2^{31}-1$;
element 22-23, initial capacity is 0;
element 24-25, incremental number is 1;
element 26, incremental percent is 0;
element 27, dynamic capacity expansion is 1.

For mode 208:

element 1, primary capacity is 0;
element 2, current capacity is current number of entries in data set;
element 3, maximum capacity is $2^{31}-1$;
element 4, expansion threshold in percent is -1 ;
element 5, delete chain head is 0;
element 6, high water mark is 0;
element 7, expansion threshold in blocks is -1 .

DBINFO modes 3xx

IMAGE

Mode 301: path info for a data set

Mode 302: key/search item for a data set

Eloquence

Mode 301: path info for a data set

Mode 302: key/search item for a data set

Mode 303: path info for a data set, returns sort item

TurboIMAGE compatibility

- Modes 301 and 302 are supported
- Mode 301 always returns all path information and mode 302 always returns the search item

© 2002 Marxmeier Software AG
EQ HP3K Training.ppt
August 9, 2002
page 85

This slide shows the TurboIMAGE 300s DBINFO modes.

Eloquence has the DBINFO modes 301 to 303. Eloquence DBINFO mode 303 is similar to mode 301 but returns the sort item in addition.

The Eloquence TurboIMAGE compatibility supports the DBINFO modes 301 and 302.

TurboIMAGE does not return a specific path if user does not have access capability to the search item. Since Eloquence does not support item level security, DBINFO mode 301 will always return all path information and mode 302 always returns the search item.

DBINFO modes 4xx

IMAGE

- Mode 401: info related to user logging
- Mode 402: info related to ILR
- Mode 403: info related to dynamic txn
- Mode 404: info related to multiple database txn
- Mode 406: database information

Eloquence

Eloquence does not have 4xx modes

TurboIMAGE compatibility

- Modes 401 and 404 and 406 are supported

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 86

This slide shows the TurboIMAGE 400s DBINFO modes.

Eloquence does not have equivalent DBINFO modes.

The Eloquence TurboIMAGE compatibility supports the DBINFO modes 401 to 404 and 406.

DBINFO modes 4xx (compatibility)

Mode 401: Eloquence does not support user logging, returns space for element 1 – 4 and 0 for the rest

Mode 402: Eloquence does not support ILR, 0 will be returned

Mode 403: most elements are 0, only a few elements meaningful

Mode 404: most elements are 0, only a few elements meaningful

Mode 406: returns up to 28 characters of database name with other info

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page 87

Since the Eloquence architecture differs from TurboIMAGE, some DBINFO modes return information which does not apply to Eloquence. In this case the Eloquence TurboIMAGE compatibility returns static values.

For mode 403:

element 8 – 9, user transaction number, contains transaction id;
element 13, database attached flag, returns 1;
element 14, dynamic transaction flag, returns transaction nesting level.

For mode 406:

element 1-14, database name truncated to 28 characters;
element 15, open mode, returns the mode used to call DBOPEN;
element 16, root file version, returns two characters 'EQ';
element 17, will return internal version number for the database schema;
element 18 will return number of data sets;
element 19 will return number of items.
element 20 will return number of index items.

DBINFO modes 5xx

<p><u>IMAGE</u></p> <p>Mode 501: subsystem access</p> <p>Mode 502: critical item update setting</p>	<p><u>Eloquence</u></p> <p>Mode 501: identify the index item</p> <p>Mode 502: describe an index item</p> <p>Mode 503: identify all index items</p> <p>Mode 504: identify all index items in a data set</p> <p>Mode 505: identifies all data sets available which contain an index item</p> <p><u>TurboIMAGE compatibility</u></p> <ul style="list-style-type: none"> • Modes 501: always return 3 • Mode 502: critical item update is always "allowed"
---	--

© 2002 Marxmeier Software AG
EQ HP3K Training.ppt August 9, 2002 page 88

This slide shows the TurboIMAGE 500s DBINFO modes.

The Eloquence 500s modes are different from TurboIMAGE and return index information instead.

The Eloquence TurboIMAGE compatibility supports the DBINFO modes 501 and 502. However some information returned is static:

- Mode 501: always return 3, indicating read/write access
- Mode 502: critical item update is always "allowed"

The Eloquence 500s modes are available as DBINFO modes 700s with the TurboIMAGE compatibility.

For mode 501:

element 1 will always return a 3, denotes read/write access

For mode 502:

element 1, critical item update flag, will always return 1, which is allowed; element 2, current CIU setting, will return 0 or 1 reflecting the call to DBCONTROL mode 5/6 (enable/disable CIU).

The slide is titled "DBINFO modes 9xx" and is divided into two main sections. The left section, titled "IMAGE", describes "Mode 901: language ID for the database". The right section, titled "Eloquence", states "No equivalent mode for language setting". Below the Eloquence section, there is a sub-section titled "TurboIMAGE compatibility" which contains a bullet point: "Modes 901 is supported and returns 0". At the bottom of the slide, there is a footer with the text "© 2002 Marxmeier Software AG", "EQ HP3K Training.ppt", "August 9, 2002", and "page 89".

DBINFO modes 9xx

IMAGE
Mode 901: language ID for the database

Eloquence
No equivalent mode for language setting

TurboIMAGE compatibility

- Modes 901 is supported and returns 0

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 89

This slide shows the TurboIMAGE 900s DBINFO modes.

Eloquence does not have equivalent DBINFO modes.

The Eloquence TurboIMAGE compatibility supports the DBINFO mode 901, however it simply returns a 0 as the database language id.

DBINFO mode 7xx (compatibility)

Eloquence specific modes

Mode 701: returns an index item number

Mode 702: describes a specific index item

Mode 703: identifies index items in database

Mode 704: identifies index items in data set

Mode 705: identifies all data sets available which contain an index item

The Eloquence TurboIMAGE compatibility DBINFO modes 701 to 705 are mapped to the equivalent Eloquence 500s DBINFO modes and return information on Eloquence indexes.

DBINFO mode 8xx (compatibility)

TPI modes

Mode 801: describes third-party product on the database

Mode 802: number of external and internal index files

Mode 803: indicates if TPI is enabled

Mode 811: access available for a third party key

Mode 812: describes a third party key

Mode 813: all items (including index item) in database

Mode 814: all items (including index item) in a specific data set

The Eloquence TurboIMAGE compatibility supports the TurboIMAGE TPI modes. The returned information describes the Eloquence indexes which can be access with the TPI modes.

DBINFO mode 801 returns the version of the Eloquence product and the Eloquence TurboIMAGE compatibility library.

For mode 801:

element 1 – 20, product name, the value is “HP Eloquence A.xx.yy”

element 21 – 25, version number, format is A.xx.yy.zz

element 26 – 29, install date and time, are 0

DBINFO mode 8xx (compatibility)

TPI modes (cont)

Mode 821: sets with index item

Mode 831: identifies index items available in a specific data set

Mode 832: identifies third party keyword search keys for a data set, always return 0

Mode 833: describes a specific third party key

Mode 834: describes all items grouped with the specified key, always return 0

Since the Eloquence indexes may differ from TPI products some DBINFO modes return information which does not apply to Eloquence indexes. In this case the Eloquence TurboIMAGE compatibility returns static values. Eloquence indexes currently support only a commonly used subset of the TPI functionality.

The Eloquence TurboIMAGE DBINFO modes 700s can be used as an alternative.

DBCNTROL modes

IMAGE

Mode 1,2: enable/disable for autodefer

Mode 5,6: enable/disable critical item update

Mode 7: enable for dynamic multiple database transaction

Mode 9,10: enable/disable for HWMPUT

Mode 13: adding/dropping/rebuilding B-tree for PRIV program

Mode 14: btreemode1 related setting for PRIV program

Mode 15,16: enable/disable the btreemode1

Mode 18,19: enable/disable to return warning for large transaction

Mode 20: temp negate the DBUTIL "only return old IMAGE limit" setting

Eloquence

No equivalent intrinsic

TurboIMAGE compatibility

- Modes 5,6: enable/disable critical item update
- Modes 15,16: enable/disable btreemode1
- Modes 1,2,7,18,19,20 are accepted but have no effect
- Modes 9,10,13,14 are not accepted

© 2002 Marxmeier Software AG
EQ HP3K Training.ppt August 9, 2002 page 93

Eloquence does not have an equivalent to the TurboIMAGE DBCNTROL intrinsic.

The Eloquence TurboIMAGE compatibility partially supports the DBCNTROL modes:

- Modes 5,6: enable/disable critical item update
- Modes 15,16: enable/disable btreemode1
- Modes 1,2,7,18,19,20 are accepted but have no effect
- Modes 9,10,13,14 are not accepted and a status error is returned

DBERROR / DBEXPLAIN

Equivalent functionality

- Error messages may be different
- After a status error was returned status elements 7 to 10 may include information on the Eloquence database call

The Eloquence DBERROR and DBEXPLAIN calls are equivalent to TurboIMAGE. Error messages may be different though.

With the Eloquence TurboIMAGE compatibility status elements 7 to 10 are different in case a status code was returned and may include information on the native Eloquence database call.

DBLOCK modes

IMAGE

- Mode 1: wait database lock
- Mode 2: no-wait database lock
- Mode 3: wait data set lock
- Mode 4: no-wait data set lock
- Mode 5: wait predicate lock
- Mode 6: no-wait predicate lock

Eloquence

- Mode 1 – 6: same as IMAGE
- Mode 11 – 16: read locks for database, data set and predicate lock

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 95

TurboIMAGE has 6 DBLOCK modes.

Eloquence DBLOCK modes 1 to 6 are the same. In addition, Eloquence supports DBLOCK modes 11 to 16 for READ level locking.

If a READ lock has been granted, concurrent READ locks are valid, but a concurrent WRITE locks are blocked.

DBLOCK modes (compatibility)

Mode 1: wait database lock

Mode 2: no-wait database lock

Mode 3: wait data set lock

Mode 4: no-wait data set lock

Mode 5: wait predicate lock

Mode 6: no-wait predicate lock

Mode 11: read only wait database lock

Mode 12: read only no-wait database lock

Mode 13: read only wait data set lock

Mode 14: read only no-wait data set lock

Mode 15: read only wait predicate lock

Mode 16: read only no-wait predicate lock

The Eloquence TurboIMAGE compatibility supports all Eloquence DBLOCK modes.

With Eloquence, a predicate DBLOCK using multiple lock descriptors is applied atomically. It either succeeds and is granted or rejected and not applied at all. There is no need to call DBUNLOCK after a predicate DBLOCK failed.

DBUNLOCK modes

IMAGE
Mode 1: unlock all the locks

Eloquence
Mode 1: unlock all locks
Mode 3: unlock the resource which was locked by calling DBLOCK mode 3/4/13/14
Mode 5: unlock the resource which was locked by calling DBLOCK mode 5/6/15/16

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 97

TurboIMAGE has DBUNLOCK mode 1 to unlock all locks.

Eloquence supports additional DBUNLOCK modes to allow selective unlocking. The same lock descriptor used with DBLOCK can be provided to DBUNLOCK and released independently of other locks.

DBUNLOCK modes (compatibility)

Mode 1: unlock all the resources owned

Modes 3, 5 allow selective unlocking

- DBUNLOCK during dynamic transaction is valid but delayed until transaction completes

The Eloquence TurboIMAGE compatibility supports the extended DBUNLOCK modes.

Eloquence does allow a DBUNLOCK during a dynamic transaction. However the lock will only be released after the transaction has finished (either committed or rolled back).

Allowing DBUNLOCK during a dynamic transaction simplifies the application design. The delayed unlock helps to avoid typical deadlock situations.

DBBEGIN / DBEND / DBMEMO modes

IMAGE/DBBEGIN

Mode 1: starts a transaction, a 'begin' record is written to IMAGE user log file

Mode 3: start a transaction involves more than one database; one 'begin' record for every database is written to IMAGE log file

Mode 4: start a transaction involves more than one databases, only one 'begin' record is written to IMAGE log file

IMAGE/DBMEMO

Mode 1: writes a 'memo' record to the log file

IMAGE/DBEND

Mode 1: ends a transaction

Mode 2: ends a transaction and flush the log buffer to the disk

Mode 3: ends a multiple database transaction started by DBBEGIN mode 3, each database has one 'end' record

Mode 4: ends a multiple database transaction started by DBBEGIN mode 4, only one log record is written to the log file

Static transactions in TurboIMAGE use DBBEGIN and DBEND to mark the beginning and the end of a static transaction. If enabled, log records are written to the TurboIMAGE user log and can be used in case of a roll-forward or roll-backward recovery.

Eloquence currently does not support static transactions and DBBEGIN, DBEND and DBMEMO have no effect. The Eloquence TurboIMAGE compatibility only validates the modes and arguments.

DBXBEGIN modes

IMAGE

Mode 1: starts a dynamic transaction for the current database

Mode 3: starts a dynamic transaction for multiple database; user has to provide a database name list

Mode 4: starts a dynamic transaction for multiple database; user can add database involved one at a time. That is only used by IMAGE/SQL

Eloquence

Mode 1: starts a transaction.

© 2002 Marxmeier Software AG EQ_HP3K_Training.ppt August 9, 2002 page 1/10

Dynamic TurboIMAGE transactions are specific to a database or a list of databases.

Transactions in Eloquence are not database specific All database modifications after beginning a transaction are part of the transaction until the the transaction ends.

Eloquence does not expose incomplete transactions to concurrent processes. Changes in an uncommitted transaction are only visible to the initiating process (transaction isolation).

Eloquence transactions can be nested and do not have a fixed limit in size.

DBXBEGIN modes (compatibility)

Mode 1: starts a transaction

Mode 3: starts a transaction

- Allows nested transactions
- Transactions cover all databases
- Transaction isolation does not expose partial transactions
- Transactions are not limited in size

The Eloquence TurboIMAGE compatibility supports the TurboIMAGE DBXBEGIN modes 1 and 3 which are mapped to the Eloquence DBBEGIN mode 1.

DBXBEGIN mode 4 is not supported since it is only used by IMAGE/SQL.

Since Eloquence transactions are not specific to a database, the base name or base name list arguments are ignored.

The status array (the second element) returns the transaction nesting level (after the DBBEGIN).

DBXEND modes

IMAGE

Mode 1: ends a dynamic transaction

Mode 2: ends a dynamic transaction and write the contents of the XM log buffer to the disk

Mode 3: ends a dynamic transaction with multiple databases involved. User has to provide the base name list with the global transaction ID in front

Mode 4: ends a dynamic transaction with multiple databases involved. This is only used by IMAGE/SQL

Eloquence

Mode 1: commit the current sub-transaction or top transaction if there are no sub-transactions

Mode 2: commit the top level transaction

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 1/12

TurboIMAGE has 4 DBXEND modes which correspond to the DBXBEGIN modes.

Eloquence supports two modes to commit a database transaction.

- Mode 1 is used to commit the current transaction level. If nested transactions are used, the current sub-transaction is merged with the upper transaction level. Otherwise the transaction is committed.
- Mode 2 allows to commit the current transaction including any sub-transactions.

DBXEND modes (compatibility)

Mode 1: ends a dynamic transaction

Mode 2: ends a dynamic transaction

Mode 3: ends a dynamic transaction

Mode 5: force end of a top level transaction

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page
1/11

The Eloquence TurboIMAGE compatibility supports the TurboIMAGE DBXEND modes 1 to 3 which are mapped to the Eloquence DBCOMMIT mode 1.

DBXEND mode 4 is not supported since it is only used by IMAGE/SQL.

In addition, the Eloquence TurboIMAGE compatibility adds the new DBXEND mode 5 which is mapped to the Eloquence DBCOMMIT mode 2 and allows to commit the current transaction including any sub-transaction.

The status array (the second element) returns the transaction nesting level (after the DBCOMMIT).

DBXUNDO modes

IMAGE	Eloquence
Mode 1: rollback the dynamic transaction	Mode 1: rollback the current transaction
Mode 3: rollback the dynamic transaction started by calling DBXBEGIN mode 3	Mode 2: rollback a given transaction where user has to provide the transaction ID
Mode 4: rollback the dynamic transaction started by calling DBXBEGIN mode 4	Mode 3: rollback to top level transaction

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 174

TurboIMAGE has 3 DBXUNDO modes which correspond to the DBXBEGIN modes.

Eloquence supports three modes to rollback a database transaction.

- Mode 1 is used to rollback the current transaction level. If nested transactions are used, the current sub-transaction is rolled back. Otherwise the transaction is rolled back.
- Mode 2 allows to specify a transaction id to roll back to a specific sub-transaction.
- Mode 3 rolls back the current transaction including any sub-transaction.

DBXUNDO modes (compatibility)

Mode 1: rollback the current transaction

Mode 3: rollback the current transaction

Mode 5: rollback to top level transaction

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page
1/14

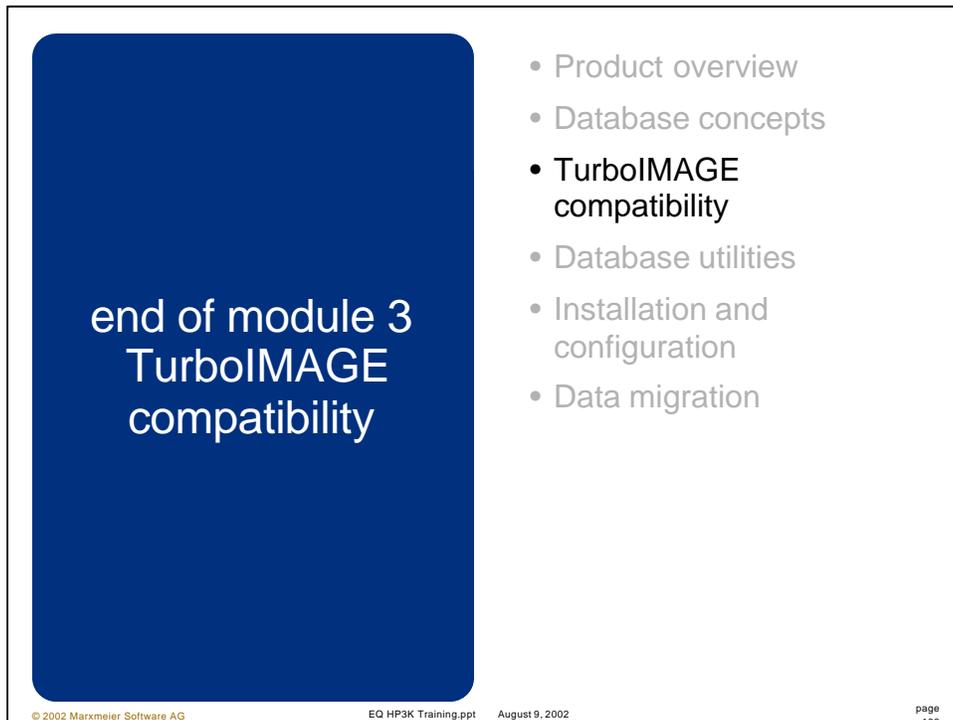
The Eloquence TurboIMAGE compatibility supports the DBXEND modes 1 and 3 which are mapped to the Eloquence DBROLLBACK mode 1.

DBXEND mode 4 is not supported since it is only used by IMAGE/SQL.

The Eloquence TurboIMAGE compatibility adds the new DBXUNDO mode 5 which is mapped to the Eloquence DBROLLBACK mode 3 and allows to rollback the current transaction including any sub-transaction.

The status array (the second element) returns the transaction nesting level (after the DBROLLBACK).

The Eloquence option to rollback to a specific sub-transaction is currently not supported.



end of module 3
TurboIMAGE
compatibility

- Product overview
- Database concepts
- TurboIMAGE compatibility
- Database utilities
- Installation and configuration
- Data migration

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 1 of 1

In this module we had a detailed look at the Eloquence TurboIMAGE compatibility.

As we have seen, most of the TurboIMAGE intrinsics have a native equivalent in Eloquence and the TurboIMAGE compatibility handles the differences.

While there are differences to TurboIMAGE in some details, most of the differences are not significant and should rarely require application changes.

module 4 database utilities

- Product overview
- Database concepts
- TurboIMAGE compatibility
- **Database utilities**
- Installation and configuration
- Data migration

The database utilities module provides an overview on the Eloquence database utilities. Eloquence provides an extensive set of database utilities which cover a wide range of activities.

We give a short introduction to each utility program and explains its typical usage.

database utilities

- Offline utilities can be used when the database server is not active
- Administrative utilities to control server operation or request server status information
- Database utilities to use the database server

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 108

The Eloquence database utilities can be separated in three categories.

- Off-line utilities which are used when the database server is not active.
- Administrative utilities to control the server operation or request request status information
- and database utilities which access the database server

offline utilities

- dbvolcreate, dbvoextend, dbvolchange, dblogreset
 - database volume management
- dbvoldump
 - display volume properties
- dbfsck
 - volume consistency check and simple repair tool
- dbrecover
 - forward recovery

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 109

The Eloquence off-line utilities operate on database volume files and can not be used when the database server is running.

Off-line utilities include dbvolcreate, dbvoextend, dbvolchange and dblogreset to perform database volume management.

dbvoldump to display volume properties.

dbfsck to check volume consistency and perform repairs and dbrecover to perform roll-forward recovery

dbvolcreate

The dbvolcreate utility is used to create a new server instance. It creates the primary data volume and initializes the server catalog

The volume file is added to the server configuration file

```
# dbvolcreate /db/d01.vol
```

Options:

-c config_file	server config file
-s initial_size	initial size of the volume file
-m max_size	max. volume file size
-e extension_size	volume extension size

The dbvolcreate utility is used to setup a new server instance and creates the primary data volume which contains the initial server catalog.

The specified volume file is created and then added to the server configuration. Option -c allows to specify a server configuration file.

By default the volume file is created with the minimum size and allowed to grow dynamically. Initial size, maximum size and the extension size can be specified through command line options.

The option -s specifies the initial size (in MB). If given the volume file is created with this size.

The option -m specifies the maximum volume size (in MB). By default the volume is allowed to grow until it reaches an internal limit, which currently is about 2 GB.

The option -e specifies the extension size (in MB). When the disk space in a volume has been exhausted the volume is extended by the extension size. If the extension size is zero, the volume is not extended automatically by the server. The default extension size is 1 MB.

This command is typically used by the system administrator (root) or executed from the account owning the database files.

dbvolextend

The dbvolextend utility extends a database environment by a volume file (data or log volume)

The volume file is added to the server configuration file

```
# dbvolextend -t log /db/d03.vol
```

Options:

-c config_file	server config file
-t volume_type	the volume type (DATA or LOG)
-s initial_size	initial size of the volume file
-m max_size	max. volume file size
-e extension_size	volume extension size

The dbvolextend utility extends a database environment by a volume file. This can either be a transaction log volume or another data volume. The volume type is specified with the option -t. Valid volume types are DATA and LOG.

The specified volume file is created and then added to the server configuration. Option -c allows to specify a server configuration file.

By default the volume file is created with the minimum size and allowed to grow dynamically. Initial size, maximum size and the extension size can be specified through command line options.

The option -s specifies the initial size (in MB). If given the volume file is created with this size.

The option -m specifies the maximum volume size (in MB). By default the volume is allowed to grow until it reaches an internal limit, which currently is about 2 GB.

The option -e specifies the extension size (in MB). When the disk space in a volume has been exhausted the volume is extended by the extension size. If the extension size is zero, the volume is not extended automatically by the server. The default extension size is 1 MB.

This command is typically used by the system administrator (root) or executed from the account owning the database files.

dbvolchange

The dbvolchange utility is used to change properties of a volume file

```
# dbvolchange -m 0 /db/d01.vol
```

Options:

-c config_file	server config file
-m max_size	max. volume file size (0 = max.)
-e extension_size	volume extension size (0 = fixed size)

The dbvolchange utility is used to change the properties of a volume file, such as the maximum size or the extension size.

Option -c allows to specify a server configuration file.

The option -m specifies the maximum volume size (in MB). If the maximum size is zero, the volume is allowed to grow until it reaches an internal limit, which currently is about 2 GB.

The option -e specifies the extension size (in MB). When the disk space in a volume has been exhausted the volume is extended by the extension size. If the extension size is zero, the volume is not extended automatically by the server.

A common use of dbvolchange is to remove the 50 MB limit from the database volume imposed by the "Personal Edition" license after installing a license key.

This command is typically used by the system administrator (root) or executed from the account owning the database files.

dblogreset

The `dblogreset` utility is used to reset the transaction log volume to minimal size

`dblogreset` processes any information in the transaction log volume before truncating the file

```
# dblogreset
```

Options:

```
-c config_file      server config file
```

The *dblogreset* utility is used to reset the transaction log volume to a minimum size.

The transaction log volume in a database environment is used to hold information on recently committed transactions. The transaction log volume(s) also provide temporary storage for incomplete transactions and during online backup.

In case the database server was not shutdown cleanly, the log volume holds important data such as a journal of the recently committed transactions or operations performed during online backup

Although disk space is freed in the log volume, the size of the volume file does not shrink. This can be achieved with the *dblogreset* utility.

`dblogreset` processes any outstanding operation in the log volume(s) before truncating the file(s).

Option `-c` allows to specify a server configuration file.

This command is typically used by the system administrator (root) or executed from the account owning the database files.

dbvoldump

The dbvoldump utility is used to display volume properties

```
# dbvoldump
```

Options:

```
-c config_file      server config file
```

The dbvoldump utility is used to output volume information (such as used and available disk space with in the volume files) when the server is not active. When the database server is running, this information can be requested from the database server.

Option -c allows to specify a server configuration file.

This command is typically used by the system administrator (root) or executed from the account owning the database files.

dbvoldump utility

```
$ dbvoldump
```

```
ID  Type Path
1   DATA /data/db/db01.vol
2   LOG  /data/db/db02.vol
3   DATA /data/db/db03.vol
4   DATA /data/db/db04.vol
5   DATA /data/db/db05.vol
6   DATA /data/db/db06.vol
```

```
ID  Type  Cur. Sz  Ext .Sz  Max. Sz  Free  Used
1   DATA  1024.0   1.0    1024.0   0.0   1024.0
2   LOG    356.5   1.0     0.0    356.4   0.1
3   DATA  1024.0   1.0    1024.0   0.0   1024.0
4   DATA  1024.0   1.0    1024.0   0.0   1024.0
5   DATA  2000.0   1.0    2000.0   0.0   2000.0
6   DATA   642.5   1.0    1024.0   3.4   639.1
```

This is an example output from the dbvoldump utility.

The output includes the volume files used by the database environment, the volume type, current size (in megabytes), extension size, maximum size and the free and used disk space for each volume.

dbfsck utility

The dbfsck utility can be used to check volume consistency and perform simple repairs

It currently does not verify IMAGE integrity

dbfsck

Options:

-c config_file	server config file
-w	enable write mode
-a	automatically fix minor inconsistencies

The dbfsck utility allows to check the consistency of the volume files and perform repairs. dbfsck does not operate at the IMAGE level but verifies the consistency of internal objects on which IMAGE relies.

dbfsck started its life as a support tool which was used to check for volume corruption in case of a server abort (for example after a system crash) and allow data recovery in emergency cases (for example in case a backup is not available). The repair abilities are limited but usually allow to bring up the server process and unload the data.

It is important to know that dbfsck ignores the content of the transaction log volume. In case the server was not shutdown cleanly recovery information is contained in the log volume and dbfsck may output false diagnostics. The dblogreset utility can be used to apply any pending changes from the log volume to the data volume(s).

Option -c allows to specify a server configuration file.

By default dbfsck operates in read-only mode. The option -w enables the repair mode. Option -a allows dbfsck to fix minor inconsistencies automatically for which a safe recovery is possible. Otherwise the user is prompted for each change.

dbfsck can take considerable time, depending on the amount of data.

This command is typically used by the system administrator (root) or executed from the account owning the database files.

dbrecover

The dbrecover utility is used to perform forward recovery
Speed depends on disk performance, only minimal
processing is involved

```
# dbrecover
```

Options:

```
-c config_file      server config file
```

The dbrecover utility is used to perform a roll-forward recovery.

If forward logging is enabled, all database changes since a previous backup are saved. In case a recovery becomes necessary (for example due to a system crash or hardware failure) a previous backup is restored and dbrecover is used to apply the recorded changes. Recovery time depends on the disk performance, only minimal processing is involved.

Option -c allows to specify a server configuration file.

This command is typically used by the system administrator (root) or executed from the account owning the database files.

administrative utilities

- dbctl
 - server management utility
- HTTP status monitor
 - access server status information with a web browser

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 118

The Eloquence administrative utilities are used to control the server operation or request request status information.

The dbctl utility is used to submit requests to the database server.

The HTTP status is built into the database server and allows to use a web browser to monitor server operation.

dbctl utility

The dbctl utility is used to submit administrative requests to the database server or obtain status information

The dbctl utility provides the communication only. The functionality is implemented by the server process

```
$ dbctl help
$ dbctl -u dba shutdown
$ dbctl -u dba backup start
$ dbctl -h lxsrv:8302 list thread
```

Options:

-u user_id	user name
-p password	password for the user
-h host	host and/or service

The dbctl utility is used to submit administrative requests to the database server process or obtain status information.

dbctl provides the communication only. The functionality is implemented by the server process and the available commands (and arguments) depend on the server.

The option -u specifies the user id and the option -p specifies the password for the user. The option -h is used to specify the server process to contact.

Example usage:

```
$ dbctl help
$ dbctl -u dba shutdown
$ dbctl -u dba backup start
$ dbctl -h lxsrv:8302 list thread
```

dbctl commands

help [command]

- returns the list of available commands or command syntax

shutdown [seconds|CANCEL]

- shutdown the server process

backup {START|STOP}

- start or stop online backup mode

dbstore database target

- store a single database to a file, device or process

dbrestore [/info] source [dbase_name]

- restore a previously stored database

The help command returns a list of operations supported by the server. If a command name is provided, additional information on the command syntax is returned.

The shutdown command is used to shutdown a database server instance. The shutdown can be scheduled at a later time. A scheduled shutdown can be canceled with the shutdown cancel command.

The backup start/stop commands are used to indicate the begin and the end of an online backup to the server process. They are typically used in a backup script.

The dbstore/dbrestore commands allow to dump/restore a single database to/from a file, device or process (pipe). This requires previous configuration in the [devices] section of the server configuration file.

dbctl commands (cont)

`list [session|db|lock|thread]`

- list active sessions, open databases, locks and internal threads

`cancelthread {tid}`

- unblock a session which is waiting for a lock

`killthread {tid}`

- terminate a session

`logfile {logfile}`

- change the log file of the server process

`logflags {logflags}`

- change the logging options of the server process

The list command request status information from the server process (similar to the HTTP status). It can return information on active sessions, opened databases, lock requests and internal thread status.

The cancelthread operation can be used by the administrator to unblock a user session which is waiting for a lock (thread in status W). The thread_id argument can be obtained by list threads. A status is returned to the application.

The killthread operation can be used by the administrator to terminate a user session. The thread_id argument can be obtained by list threads. Status -700 is returned to the application.

The logfile and logflags commands allow to change the server log file and logging mode.

HTTP status

The HTTP status monitor is a simple web server built into the database server

A web browser can be used to check the database server status

Enter the address in your web browser <http://server:8103/> where 8103 is the port number configured for HttpService

The dbctl utility can be used to retrieve similar information from a script file

The HTTP status monitor is a simple web server built into the Eloquence database server. If enabled it allows to use a web browser to access server status information.

The HTTP status is enabled in the server configuration file by configuring a port number (or service name) for the ServiceHttp configuration item.

```
[Server]
ServiceHttp = 8103
```

To access the HTTP status information enter the server hostname and the port number in your web browser <http://server:8103/>

- where server is is the IP address or hostname of your server system
- and 8103 is the port number configured for ServiceHttp.

Related configuration items:

HttpUser If set, a matching user name is required

HttpPswd If set, a matching password is required

HttpFrame If set, the status information can be embedded in a frame

HTTP status

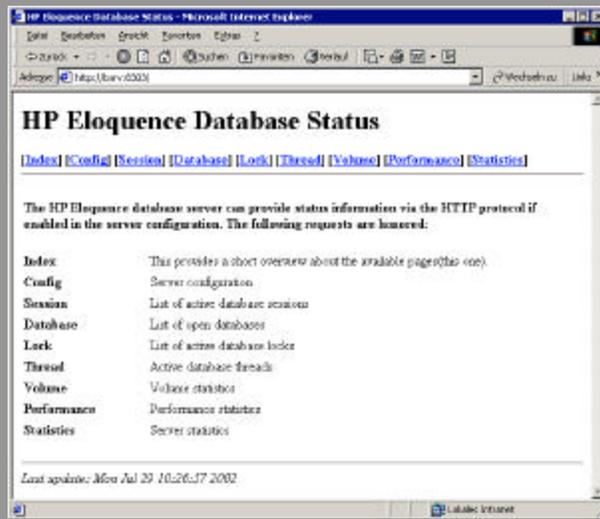
- Server configuration
- Active database sessions
- Opened databases
- Database locks
- Status of the server threads
- Volume file usage and status
- Performance information
- Internal statistics

The following information is available through the HTTP status

- The server configuration
- Active database sessions
- Opened databases
- Database locks
- Status of the server threads
- Volume file usage and status
- Performance information
- Internal statistics

Each section is displayed in a separate web document. We will look at a few examples in the next slides.

HTTP status (index)



The config page returns the server configuration.

HTTP status (config)

Server Status

Item	Value
Server revision	@(#)HP-B1368B A.06.31 HP ELOQUENCE / Linux (Jul 7 2001)
Server process id	32008
Semaphore id	40962
Installed licenses	Permanent, <i>unlimited</i>

Server Configuration

Configuration Item	Setting
panic	restart
service	eloqdb
serviceHTTP	8103
UseKeepAlive	Yes
NoDNS	No

The config page outputs the server configuration.

HTTP status (session)

Active Sessions

Available number of user licenses: unlimited
Number of active sessions: 3

Session	TID	Host	IP ADDR	Platform	User	Login
0x827d6d0	8	bxmike.marxmeier.com	194.64.71.106:2309	Linux	mike	public
0x82498c8	9	localhost	127.0.0.1:2822	Linux	mike	public
0x82552d0	10	localhost	127.0.0.1:2817	Linux	rolf	public

The session page returns information the current database sessions. For each session, the following information is returned:

Session	- Internal session id
TID	- Internal thread id
HOST	- Host name of the client system
IP ADDR	- IP address of the client system
Platform	- Client operating system
User	- User login on the client system
Login	- Database login

HTTP status (database)

Open Databases

Database	OpenCount	WriteShared	WriteExclusive	ReadShared	ReadExclusive
AHDB	1	1	0	0	0
VHDB	1	1	0	0	0
BHDB	1	1	0	0	0
SAMPLE	1	1	0	0	0
PPSSC	1	1	0	0	0
KALK5	1	1	0	0	0
ZEWI5	1	1	0	0	0

The database page returns a list of the currently opened databases and how often they are opened.

HTTP status (lock)

Database Locks

Session	Database	DBID	Status	Mode	Qualifier
0x827d6d0	SAMPLE	1	GRANTED	6	expression
set=1 item=@ (effective set lock)					
0x825a440	SAMPLE	1	BLOCKED	4	set 1

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

Page 108

The lock page returns the currently active the database locks. Both granted and blocked lock requests are displayed.

For each lock, the following information is returned:

Session	- Internal session id
Database	- Name or the database
DBID	- Internal database id
Status	- GRANTED or BLOCKED
Mode	- Lock type, equivalent to DBLOCK modes Mode 2,12: Database lock (Read/Write, Read) Mode 4,14: Data set lock (Write/Read, Read) Mode 6,16: Predicate lock (Write/Read, Read)
Qualifier	- Lock qualifier

In case of a predicate lock, the lock descriptor is displayed in a separate line.

HTTP status (threads)

Active Threads

Number of active user threads: 4

ST	TID	WCHAN	Blocked	Stack	Avail	Sched	Name
R	1			120	6872	862908	idle
S	2		262552	152	16196	1	shutdown
S	3		262552	128	16220	1	ps
S	4		2	128	15120	53192	syncer
R	5			208	13536	67	http
S	6		113	128	16168	8233	tio monitor
S	7		52	140	7780	4434	checkpoint
S	8		49	140	7100	29	bmike.marxmeier.com
S	9		197	140	7744	9	localhost
S	10		266	140	7744	12	localhost
W	11	0x827cd44	3	444	7744	5	localhost

file lockdb.c, line 320 waiting for 8

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

Page
100

The thread page returns the internal thread status. For each thread, the following information is returned:

- ST - Thread status
 - R: Runnable
 - S: Suspended/Idle
 - W: Blocked (interruptable)
- TID - Internal Thread Id
- WCHAN - Internal object the thread blocks on
- Stack - Currently used stack size
- Avail - Available stack space (high mark)
- Blocked - How long the thread has been blocked (seconds)
- Sched - How often the thread has been run
- Name - Thread name (internal name or host name)

If a thread is blocked additional information on the blocking reason and dependency on another thread is returned (for support usage).

HTTP status (volume)

Server Volumes

ID	Type	Path
1	DATA	/data3/dalex/db/db01.vol
2	LOG	/data3/dalex/db/db02.vol
3	DATA	/data3/dalex/db/db03.vol
4	DATA	/data3/dalex/db/db04.vol
5	DATA	/data3/dalex/db/db05.vol
6	DATA	/data3/dalex/db/db06.vol
7	DATA	/data3/dalex/db/db07.vol

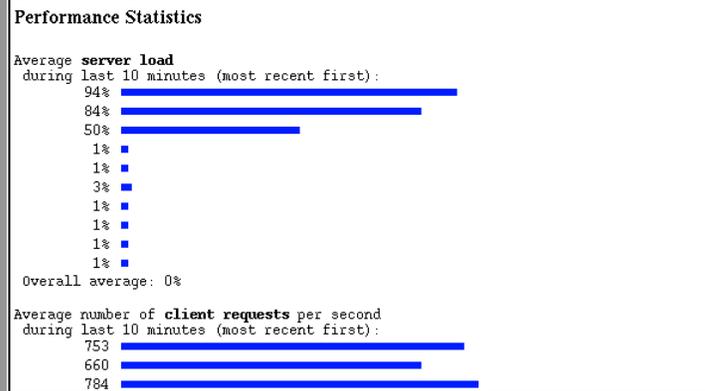
Volume Statistics

ID	Type	Cur.Sz	Ext.Sz	Max.Sz	Free	Used	Seek Cnt	Read Cnt	Write Cnt
1	DATA	1024.0	1.0	1024.0	0.0	1024.0	285	3251	277
2	LOG	48.5	1.0	0.0	48.4	0.1	27777	14	30279
3	DATA	1024.0	1.0	1024.0	0.0	1024.0	9	1237	6
4	DATA	1024.0	1.0	1024.0	0.0	1024.0	9	418	6
5	DATA	1024.0	1.0	1024.0	0.0	1024.0	34	1188	28
6	DATA	1024.0	1.0	1024.0	0.0	1024.0	7	1215	6
7	DATA	1024.0	1.0	0.0	931.3	92.7	2433	853	2568

The volume page returns volume information. Besides the volume file name, the following information is returned:

- ID - Volume Index
- Type - Volume file type
- Path - Path to volume file
- Cur.Sz - Current size of volume file (MB)
- Ext.Sz - Extension size for this Volume file (MB)
- Max.Sz - Maximum size for the volume file (MB)
- Free - Available space in volume file (MB)
- Used - Used space in volume file (MB)
- Seek Cnt - Number of repositions in this volume
- Read Cnt - Number of read accesses to this volume
- Write Cnt - Number of write accesses to this volume

HTTP status (performance)



The performance page returns information on the server usage within the last 10 minutes:

Server Load

- Percentage of time spent serving client requests vs. waiting on new requests. A low server load indicates that the server is waiting for the clients most of the time.

Client requests

- Number of client requests per second.

Committed transactions

- Number of committed transactions per second.

Disk read/write/sync requests

- Number of disk read, write or sync requests per second.

HTTP status (statistics)

Transaction Statistics

```
Number of top-level transactions: 123  
Number of sub transactions: 1655  
Number of transactions committed: 1778  
Number of transactions rolled-back: 0  
Number of transaction records in use: 0 (0 bytes)  
(maximum: 111 records, 4440 bytes)
```

Page Cache Statistics

```
18688130 page puts, 18683190 page gets, 15135 page new  
6658 page allocs, 3242 page flushes  
100% cache hit rate (18674713 hits, 8477 misses)  
75% page hash efficiency (14381 primary, 4680 synonym)  
8477 page reads, 30465 page writes
```

BTREE Statistics

```
0 key cache hits, 0 misses  
6 page splits (4 root splits, 0 sort splits)  
Key optimization reduced 0 bytes on node pages.
```

The statistics page returns internal statistics and measured disk access times.

database utilities

- schema
- dbcreate, dberase, dbpurge
 - create / erase / purge a database
- dbtables
 - database structure information
- prschema
 - create schema file from database
- dbdumpcat
 - catalog information utility

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 133

Eloquence comes with a number of database utilities

The schema processor is used to create a new database from a schema file

The dbcreate/dberase/dbpurge utilities are used to create, erase or purge a database or a part thereof

The dbtables utility returns information on the database structure

The prschema creates a schema file from a database

The dbdumcat utility allows access to catalog information

database utilities
(cont)

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

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 134

The dbexport and dbimport utilities are used to export or import a database to or from a text file.

The dbinfo utility provides an overview on the data sets in a database.

The dbutil utility is used for security and structural maintenance of a database

QUERY allows access to the database content

schema processor

The schema processor creates the database from a schema file

\$ schema -T schemafile

- The Eloquence schema processor understands both Eloquence and TurboIMAGE schema files
- Option -T selects TurboIMAGE compatibility mode
- \$CONTROL options and the database name can be specified on the command line

The schema processor creates a database from the schema file

Eloquence does not use ROOT files. The schema processor creates a separate name space in the server catalog and uploads the database structural information.

The Eloquence schema processor allows to specify most schema options on the command line (as an alternative to using \$CONTROL directives). Options specified on the command line take precedence over \$CONTROL settings.

The database name can be specified on the command line and takes precedence on the DATA BASE specification in the schema file.

The Eloquence schema processor understands both Eloquence and TurboIMAGE schema files.

The option -T is used to specify TurboIMAGE compatibility mode. In this case, the schema processor handles TurboIMAGE schema files.

A common mistake is to omit the -T command line option with a TurboIMAGE schema file. Since Eloquence schema files are similar, the schema processor may succeed but the resulting database may be incompatible.

schema processor (options)

Usage: schema [options] file

options:

```
-help      - show usage (this list)
-u user    - set user name
-p pswd    - set password
-h host    - host and service to contact
-b name    - specify database name
-l         - output source listing          (LIST)
-n         - no root file, only check syntax (NOROOT)
-t         - output set table              (TABLE)
-e cnt     - Abort after cnt error messages (ERRORS=)
-T         - HP3000 TurboImage compatibility mode
-W width   - Limit line length to width
-L         - Add line number to source listing
```

This shows the schema command line options.

The option -u is used to specify the user name and the option -p provides the password for the user.

The option -h is used to specify the server process to contact.

The option -T enables TurboIMAGE compatibility mode.

dbcreate / dberase / dbpurge

The dbcreate, dberase and dbpurge utilities are used to create, erase or purge a database

When erasing a detail set (but not the associated master set) the path information in the master set is reset

```
$ dbcreate database  
$ dberase database [set ...]  
$ dbpurge database
```

Options:

```
-u user_id      user name  
-p password    password for the user
```

prschema utility

The prschema utility is used to create a schema file from a database

If option -T is specified the schema file uses TurboIMAGE conventions where possible

```
$ prschema database
```

Options:

-u user_id	user name
-p password	password for the user
-T	TurboIMAGE compatibility

dbtables utility

The dbtables utility is used to output information on the database structure (items, data sets, paths, entries and indexes)

```
$ dbtables database
```

Options:

-u user_id	user name
-p password	password for the user
-l lines	set page length
-s	short listing (data sets only)
-T	TurboIMAGE compatibility

If the command line option -T is specified the item types will be reported in a TurboIMAGE compatible manner.

If the -s option is specified, only entry and item information is returned.

dbtables utility (cont)

DATA SET FIELDS

Data Set Name	Set Num	Item Name	Item Num	Item Type	Start Byte	Item Cnt	Item Len	Item Role
CUSTOMERS	1	CUSTNO	1	X6	0	1	6	Srch
		NAME1	3	X32	6	1	32	
		NAME2	4	X32	38	1	32	
		NAME3	5	X32	70	1	32	
		MATCHCODE	2	X10	102	1	10	
		STREET	6	X32	112	1	32	
		ZIPCITY	7	X32	144	1	32	
		PHONE	8	X18	176	1	18	
		TURNOVER	9	E4	194	3	24	
		SALESAREA	10	X6	218	1	6	
	IMATCHCODE	34				1	10	Indx
	MATCHCODE	2	X		0		10	Seg

dbdumpcat utility

The dbdumpcat utility returns information from the server catalog

- Unless a database name is specified, dbdumpcat returns server specific information
- If a database name is present, database specific information is returned
- The option -l provides a list of available catalog tables
- The -t <id> option allows to specify to return information from a specific table only. The id can be obtained from the output return by the -l option.

The dbdumpcat utility returns information from the the server catalog.

The dbdumpcat utility initially was intended as a support tool. However some catalog information is useful, even if you are not interested in the details. For example, the catalog table sysdb returns the list of all databases managed by a server instance.

Unless a database name is specified, dbdumpcat returns server specific information. If a database name is present, database specific information is returned.

The option -l provides a list of available catalog tables instead of the table contents. The -t <id> option allows to specify to return information from a specific table only. The id can be obtained from the output return by the -l option. The option -n returns the information in an easily scriptable format.

dbdumpcat utility (cont)

```
$ dbdumpcat -t 31
```

```
-----  
#31 sysdb (49 entries)  
-----
```

dbid	name	flags	nodeid
12	PPS5C	04000000	128
15	PLAN5	04000000	576
18	ZEWI5	04000000	630
21	KALK5	04000000	846
28	SAMPLE	04000000	1414
75	RUECKM	04000000	2661
82	LVS5C	04000000	2794
86	SHDB	04000000	2879
87	ATES2	04000000	3168
...			

The example shows the content of catalog table id 31 (sysdb) which holds the list of database maintained by this server.

dbexport utility

The dbexport utility is used to export the database content to text format

By default a separate file is created for each data set in the current directory named like DBNAME.###.exp

- The `-o` option allows to specify a different target directory
- The `-s file` option specifies a single file should be used which contains all data sets
- The `-c` option exports entries in chain order
- Any data set name(s) or number(s) specified after the database name indicates only the specified data sets should be exported

The dbexport utility is used to export the database content into text format. dbexport performs a sequential or chained read of the database.

By default a separate file is created per data set. Filenames of export files include the database name and the data set number. No export file created for automatic or empty data sets.

Export files are created in the the current directory. The option `-o` can be used to specify a different target directory.

If the option `-s` is specified all output will be written to a single file or sent to another process (`'-'` specifies stdout). If a hyphen is given as the output file, stdout is used.

By default dbexport performs a serial DBGET on each data set. The option `-c` exports the entries in chain order. In this case, the master set of the primary path defines the export order.

dbexport utility (options)

```
usage: dbexport [options] database [set ...]
options:
  -help    - show usage (this list)
  -u user  - set user name
  -p pswd  - set password
  -o path  - set output directory (not single file)
  -v       - verbose output
  -c       - chained export
  -r       - create restructure information
  -s file  - output into single file, '-' = stdout
  -f sep   - field separator, default is ','
  -z cset  - set export code set (roman8, iso88591)
```

This screen shows the usage of dbexport and the available options.

You need to specify a database. Data set names or numbers can be specified after the database name. If present only the given data sets are processed. By default, all data sets are exported.

Any options are specified before the database name. The option -u specifies the user id and option -p allows to specify the password for the user.

The option -v displays the export progress.

If the -r option is specified additional information on item names and item types is embedded in the export file.

dbexport utility (cont)

```
$ dbexport -v TESTDB
```

```
Processing database : TESTDB
```

```
Export path       : .
```

DATA SET		RECORDS	COUNT
CUSTOMERS	001 M	1177	1177
PARTS	002 M	182	182
ID	003 A	47	
ORDERS	004 D	47	47
LINEITEMS	005 D	136	136

This shows the output of dbexport when the -v option is given.

The output includes the data set, number of entries and number of records processed.

dbimport utility

The dbimport utility is used to load the database from export files

By default dbimport expects the export files in the current directory with a name like DBNAME.###.exp

- The -i option allows to specify a different source directory
- The -s *file* option specifies a single file should be used which contains all data sets
- The -v option displays the progress
- Any data set name(s) or number(s) specified after the database name indicates only the specified data sets should be imported

The dbimport utility allows to import a database from export files. By default a separate file is used per data set, which is expected in the current directory. Filenames of export files include the database name and the data set number. The option -i allows to specified a different directory.

If the -s option is specified a single file is used which could contain multiple data sets ('-' specifies stdin).

The dbimport utility converts the text file from text format into the database entry and performs a DBPUT.

dbimport utility (options)

```
usage: dbimport [options] database [set ...]
options:
  -help      - show usage (this list)
  -u user    - set user name
  -p pswd    - set password
  -i path    - set import path
  -v         - verbose output
  -t         - trace item value assignment
  -r file    - restructure database, '-' = no file
  -s file    - import from single file, '-' = stdin
  -e file    - log errors instead of aborting
  -f sep     - field separator, default is ','
  -z cset    - set import code set (roman8, iso88591)
```

This screen shows the usage of the dbimport utility.

You need to specify a database. Data set names or numbers can be specified after the database name. If present only the given data sets are processed.

Any options are specified before the database name. The option -u specifies the user id and option -p allows to specify the password for the user.

The option -v displays the import progress.

The option -z can be used to specify the character set encoding used in the export file. On HP-UX, by default the hp-roman8 encoding is used, Linux and Windows use the iso-8859-1 encoding.

The option -e allows dbimport to continue in case some database errors are encountered. This includes a missing chain head or duplicate entries. In this case, the error and the data are added to the log file.

The option -r allows to specify that item values are assigned based on the item name, rather than the position in the export file. This requires a previous dbexport -r to include the item names in the export file. This can be used if the source and the target database are different. A restructure file can be used to override the default item value assignments. If not present, a hyphen can be specified which indicates to use the default behavior.

EXPORT file format

- Export files are text based
 - Each line contains a record
 - Item values are separated by comma
 - String items are enclosed in quotes
- Format documentation
<http://www.hp-eloquence.com/support/misc/export.html>

Export files are text based and use a simple format. Each line contains a single record, item values are separated by a comma and string items are enclosed in quotes

The file format is documented on the Eloquence web site.

EXPORT file format (cont)

```
"24601",442,900126,"21089",5,"VK",1298.46
"24602",1120,880116,"25001",5,"VK",28073.01
"24603",1210,880125,"30010",5,"VK",1611.09
"24604",1258,880201,"13005",5,"VK",10508.16
"24605",1446,880227,"13007",11,"VK",0
"24606",1460,880227,"17007",7,"VK",1150.81
"24607",1462,880227,"17007",7,"VK",8300.82
"24608",2424,880704,"22002",5,"VK",3719.23
"24609",2612,880725,"22016",5,"VK",1396.02
"24610",2894,880907,"18012",5,"VK",14.56
"24611",3342,881027,"15017",5,"VK",808.33
...
```

This screen shows an example export file.

You can see that items are separated by a comma and string values are enclosed in quotes.

Depending on the options specified, additional information is embedded in the file.

If the option `-r` (restructure) is used with `dbexport`, item names and type information is included in the file and can be used with `dbimport`.

If the option `-s` (single file) is specified with `dbexport` all data sets are exported in a single file (or device) and the export file includes additional lines to separate between different data sets.

dbinfo utility

dbinfo lists the data sets for the specified database
data set name, type, number of entries and capacity are
output

```
$ dbinfo database [set ...]
```

Options:

```
-u user_id      user name  
-p password    password for the user
```

The dbinfo utility lists the accessible data sets for the specified database.

Data set names or numbers can be specified after the database name. If present only the given data sets are output.

Any options are specified before the database name. The option -u specifies the user id and option -p allows to specify the password for the user.

The dbinfo utility basically executes DBINFO mode 202 and displays the information. This allows easily to get an overview on a database and can also be used in scripts.

Please keep in mind that with Eloquence the capacity value specifies the highest record number ever used for this data set.

dbinfo utility

```
$ dbinfo TESTDB
Processing database : TESTDB
```

SET NAME		RECLEN	CAPACITY	ENTRIES
CUSTOMERS	001 M	112	1355	1177
PARTS	002 M	53	524	182
ID	003 A	2	2259	47
ORDERS	004 D	21	1008	47
LINEITEMS	005 D	22	1008	136

This screen shows a typical dbinfo output.

The data sets, the record length, capacity and number of current entries are displayed

dbutil utility

dbutil provides a central point for database administration

- security management
 - adding and deleting user accounts
 - granting access privileges
- structural maintenance
 - adding an item or an index to a database
 - changing an item type
 - adding a data set to a database

The dbutil utility is used to manage various aspects of a database.

The most important operations are

- security management
 - for example, adding and deleting user accounts
 - and granting access privileges
- and structural maintenance
 - for example, adding an item or an index to a database
 - changing an item type
 - and adding a data set to a database

dbutil utility

usage: dbutil [options] [file]

options:

- help = show usage (this list)
- i = interactive mode
- n = check script (batch mode only)
- v = verbose (batch mode only)

If a file argument is present, dbutil will process in batch mode unless the -i option is present

If the -n option is present, no changes will be made to the database. Processing will end after checking the input file and analyzing the changes

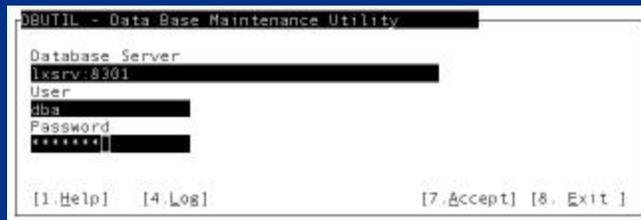
This shows the the most important dbutil command line options.

If the file argument is not present, dbutil runs in interactive mode using a dialog based user interface. dbutil can use the Eloquence JDLG component to present a graphical user interface.

If the file argument is present, dbutil is running in batch mode, executing the directives in the script file.

If the -n option is present, no changes will be made to the database. Processing will end after checking the input file and analyzing the changes.

dbutil utility



```
DBUTIL - Data Base Maintenance Utility
Database Server
lxsrv:8301
User
dba
Password
*****
[1. Help] [4. Log] [7. Accept] [8. Exit]
```

```
# connect to server
CONNECT "lxsrv:8302";
LOGON "dba" PASSWORD "file:/root/dba";

# create user mike with the password "secret"
# and allow connection to the database server
CREATE USER "mike" PASSWORD "secret";
GRANT CONNECT TO "mike";
```

This screen demonstrates the two alternatives to use dbutil.

When started in interactive mode a dialog based user interface is provided. An example dialog is shown in the upper example.

Otherwise a script file can be specified which is then processed by dbutil. An example script is shown below.

QUERY utility

The Eloquence QUERY utility is different from the e3000 QUERY

- limited capabilities
- uses different syntax
- supported data types are limited
- a TurboIMAGE compatible QUERY version will be added to Eloquence

Eloquence comes with a QUERY application. However the Eloquence QUERY utility is different from QUERY on the e3000.

The Eloquence QUERY version has limited abilities and uses a different syntax. This QUERY version initially originated on the HP260 and was ported to Eloquence. Since it is rarely used not much effort was spent to improve it.

We recognize the necessity to support the thousands of custom QUERY reports currently in use and plan to include an e3000 compatible QUERY version with Eloquence.

end of module 4 database utilities

- Product overview
- Database concepts
- TurboIMAGE compatibility
- **Database utilities**
- Installation and configuration
- Data migration

This module on eloquence database utilities provided a short overview on the database utilities available with eloquence.

module 5 installation and configuration

- Product overview
- Database concepts
- TurboIMAGE compatibility
- Database utilities
- **Installation and configuration**
- Data migration

This module describes the installation and configuration of the eloquence database. We focus on the HPiUX platform. However Linux and Windows platform are covered as well.

installation and configuration

- Install the product
- Configure the operating system
- Configure automatic server startup
- Create the database environment
- Platform differences

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 1 of 8

We show how to to install the eloquence product and configure the operating system. We explain how to configure automatic server startup and how to setup the database environment. Finally we look into differences on the Linux and windows platform.

Obtaining the product

- Download from the Eloquence web site
<http://www.hp-eloquence.com/download>
- Order a CD-ROM
- Check the Eloquence web site for recent patches
<http://www.hp-eloquence.com/support>

The Eloquence product is available for download from the Eloquence web site. This is the preferred distribution method. A separate product archive is available for each platform which is only a few megabytes in size.

A CD-ROM media can be ordered as an alternative

Please check the Eloquence web site for any recent product patches.

Installation on HP-UX

The download version of the product archive must be uncompressed before installation

```
# gzip -d B1368B-A0700.sd.gz
```

Run "swinstall" to install the product

```
# /usr/sbin/swinstall -s /path/to/B1368B-A0700.sd
```

- the -s option specifies the absolute path of the product archive

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page
120

The download version of the HP-UX product archive is compressed with gzip (GNU zip) and must be uncompressed before installation. This is indicated by the .gz file extension.

To uncompress the product archive execute the gzip command as shown

```
gzip -d B1368B-A0700.sd.gz
```

This replaces the compressed archive with the uncompressed version.

Then run swinstall to install the product. The option -s can be given to swinstall to specify the absolute path of the product archive. This avoids to select the installation source interactively in swinstall.

Installation on HP-UX

Mark the product bundle for installation

- B1368B: install the base product
- B1368B-TLIC: install a 30 days temporary license in addition

Select Actions -> Install to continue the installation

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page
161

By default the product bundles are displayed in swinstall.

You have the option to install a temporary license key in addition to the product. The temporary license key is valid for 30 days and is only created once per system.

By default the Personal Edition license key is installed which imposes a two user and 50 MB database limit.

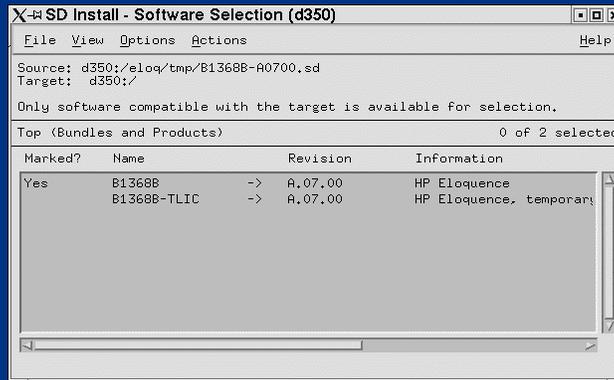
Select the option and then continue with the installation.

From the product view in swinstall you can select the tlic option to install a temporary license key.

In swinstall, select the product bundle to install (press the space bar to select) and choose Actions -> Mark for Install from the menu (the tab key selects the menu bar). A “yes” will be displayed in the “marked?” column.

Then select Actions -> Install (analysis) from the menu to start the installation. When the analysis is finished with no error, choose “Yes” in the Confirmation window to begin the actual installation process. When the installation is completed, a dialog is displayed to notify you. You may exit then.

Installation on HP-UX



This hardcopy shows the Motif version of swinstall with the main Eloquence options.

The first option to install the base product is marked for installation. The option to install a temporary license key is not marked.

Install HP-UX patches

Required HP-UX patches are documented on the Eloquence web site.

HP-UX patches can be obtained from the hp itrc site

- <http://europe.itrc.hp.com> (Europe)
- <http://us.itrc.hp.com> (US, Asia-Pacific)

Currently required HP-UX patches are:

- HP-UX 11.0
 - PHKL_24005 or later
- HP-UX 10.20
 - PHCO_10947 or later

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page 123

Required HP-UX patches are documented on the Eloquence web site. Please make sure the required HP-UX patches are installed.

HP-UX patches can be downloaded from the hp it resource center, the URLs for the European and the US-Pacific sites are shown on the slide.

Currently the HP-UX patches mentioned on the slide are required.

- HP-UX 11.0
 - PHKL_24005 or later
- HP-UX 10.20
 - PHCO_10947 or later

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
<http://www.hp-eloquence.com/license>

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page
164

Eloquence requires a license key to be installed on the server system. To evaluate the Eloquence product, the following options are available:

- By default, the Eloquence “Personal Edition” license key is installed. The Eloquence “Personal Edition” provides a free and non-expiring two user license and is limited to a database size of 50 MB. The Eloquence “Personal Edition” license is intended for product evaluation and personal usage.
- During the installation process, a temporary license key can be created (once per system) which is valid for a period of 30 days and which does not restrict the number of users or database size (option not available on the Linux platform).
- A temporary license key can be requested from the Eloquence web site by filling out the temporary key request form
<http://www.hp-eloquence.com/license/demo.html>

Install new License Key

- Add supplied license keys to the Eloquence license file
`/etc/opt/eloquence6/license`
- Comment out any existing license keys
- Use the `chklic` utility to verify the license file
`/opt/eloquence6/etc/chklic`

To install an Eloquence license key, perform the following steps:

- Add the supplied license keys to the Eloquence license file
`/etc/opt/eloquence6/license`.
- It is recommended that you comment out any unused license keys by putting a hash sign in front.
- The `chklic` utility can be used to verify the license file
`/opt/eloquence6/etc/chklic`

If a temporary license key was created during installation you may skip this step for now and add the permanent license later.

Configuration of TCP Services

Installation adds the default Eloquence services in the services file.

- On HP-UX and Linux the /etc/services file is used to define TCP service names
- On Windows the services file is located in the directory C:\Windows\System32\Drivers\etc

Make sure the eloqdb service name is defined in the services file

```
eloqdb 8102/tcp # Eloquence database server
```

The services file is used with TCP to associate a service name with a port number. It allows to use a name for a network service.

The installation process adds the default Eloquence service names in the services file.

The location of the services file depends on the operating system.

- On HP-UX and Linux the /etc/services file is used to define TCP service names.
- On Windows the services file is located in the directory C:\Windows\System32\Drivers\etc.

Make sure the eloqdb service name is defined in the services file and the port number associated with it is not already in use.

Create eloqdb user/group

- A separate user id and group are used with the database environment
 - configured in the eloqdb6.cfg
 - owner of the database volume files
 - account can be used for volume administration
- Create a user name and a group name
e.g. eloqdb
- On Windows the system account is used by default

A separate user id and group are used with the database environment.

Create a user and group, for example eloqdb.

This user and group own the database volume files and are used to run the database server.

Administration of volume files should be performed either by the system administrator (root) or use this account. If the user account is not used for volume administration it should be disabled.

On Windows, the SYSTEM account is typically used to run the database service and this step is optional.

HP-UX Configuration Kernel Parameters

If shared memory communication is enabled (EnableIPC set) Eloquence needs additional kernel resources

HP-UX kernel parameters may need to be configured

- semaphores related parameters
- shared memory related parameters
- data size

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page
128

Depending on the configuration, Eloquence has specific requirements for the kernel configuration.

The HP-UX kernel parameters must be verified if they are sufficient to run the Eloquence database in the intended configuration.

If the shared memory communication is enabled, Eloquence needs additional kernel resources.

To configure kernel parameters, start SAM and select Kernel Configuration -> Configurable Parameters, change the kernel parameters as necessary and build a new kernel.

Changing kernel parameters requires a system reboot.

HP-UX Configuration Kernel Parameters

- Semaphores related kernel parameters
 - 'semmni' : number of connections (EnableIPC)
 - 'semmap' : set to 'semmni' + 2
 - 'semmns' : number of connections (EnableIPC)
+ number i/o threads
 - 'semmnu' : number of connections (EnableIPC)
 - 'semume' : number of connections (EnableIPC)

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page
120

If shared memory communication is enabled in the database server configuration file, the kernel SYSV semaphores related kernel parameters likely need to be changed.

For each local connection, a separate semaphore is needed. The semmni, semmap, semmns, semmnu and semume kernel parameters must be adapted accordingly.

If shared memory communication is not enabled, the semmns kernel parameter needs only be changed if a large number of i/o threads is used. The default number of 4 i/o threads is likely covered by the default kernel configuration.

In the following sections the variable x specifies the number of concurrent connections (Threads configuration item) and y the number of i/o threads (IOThreads configuration item)

Semaphore configuration

- 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 $x+20$

HP-UX Configuration Kernel Parameters

- Shared memory related parameters
 - 'shmmni' : number of connections (EnableIPC)
 - 'shmseg' : number of connections (EnableIPC)
- Max. data size
 - set 'maxdsiz' to at least 0x08000000 (128MB)

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page
170

If shared memory communication is enabled in the database server configuration file, the kernel SYSV shared memory related kernel parameters need to be changed.

For each local connection, a separate shared memory segment is needed. The shmmni and shmseg kernel parameters must be adapted accordingly.

If shared memory communication is not enabled, the shared memory related kernel parameters do not need to be changed .

In any case, the maxdsiz kernel parameter should be changed to allow at least 128 MB of process data size.

In the following sections the variable *x* specifies the number of concurrent connections (Threads configuration item) and *y* the number of i/o threads (IOThreads configuration item)

Shared memory configuration

- 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)

HP-UX Configuration Automatic server start

Modify the startup configuration file
`/etc/rc.config.d/eloquence6`

```
START_ELOQSD=0  
START_ELOQDB6=1  
START_ELOQDB5=0
```

When the system is booted, the database server can be started automatically. Which Eloquence services are started is defined in a configuration file.

To automatically start the database server, set the `START_ELOQDB6` variable to one. The other variables should be set to zero, since they enable Eloquence services which are not related to the database and are not needed.

The configuration settings shown here enable the automatic start of the Eloquence database and disable other services which are not related to the database.

setup the database environment

- Edit the server configuration file
- Create the database volumes
- Start the database server

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 172

Now that the Eloquence product is installed the next step is to setup the database environment.

This step involves the following tasks

- edit the server configuration file
- and create the volume files
- then start the database server process

Database Environment

Database environment (server instance) consists of

- Server configuration file (eloqdb6.cfg)
- Primary data volume
- Transaction log volume
- Additional data volume(s)

The database environment consists of the following elements:

- The server configuration file
- The primary data volume
- A transaction log volume
- and additional data volumes as needed to provide additional disk space

Server configuration file

This file defines server properties

- configuration
- scaling and tuning parameters
- volume files

Default server configuration file is
`/etc/opt/eloquence6/eloqdb6.cfg`

The server configuration file is a text file, similar to INI file format, which specifies the server configuration, scaling and tuning parameters and the location of the volume files. It is read when the server is started.

A leading hash sign indicates a comment. The file has multiple sections with the section names enclosed in square brackets.

Example Server Configuration

```
[Server]
Service = eloqdb
ServiceHTTP = 8103
UID = eloqdb
GID = eloqdb
EnableIPC = 1
SyncMode = 0
```

The **[Server]** section defines general server settings.

The **Service** configuration item specifies the service name (as defined in the services file) or the port number which is used by the database server. The default value is `eloqdb`. The port number must be different for each server instance and may not already be used.

The **ServiceHttp** configuration item specifies the service name (as defined in the services file) or the port number used by the embedded web server. By default the HTTP status is disabled.

The **Uid** and **Gid** configuration items specify the user and group which are used to run the server and own the volume files. This setting is ignored on Windows.

The **EnableIPC** configuration item activates the use of shared memory for communication. This significantly reduces system overhead but requires additional kernel configuration.

The **SyncMode** configuration item specifies if the server should write committed transactions to disk immediately. If disabled, a more efficient write strategy is used. `SyncMode` is enabled by default which makes the database resistant in case of a system crash, however it causes additional disk load. A more efficient option is to enable forward logging.

Example Server Configuration

```
[Config]
Threads = 100
IOThreads = 4
BufferCache = 64
CheckPtSize = 20
```

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page
176

The **[Config]** section defines the scaling and tuning parameter settings for the database server.

This example configuration supports 100 concurrent users using a 64MB cache. You should at least have 128 MB memory on your system.

The **Threads** configuration item specifies the maximum number of connections. Each connection is associated with a thread running in the server context. Only a single connection is used by an application to access any number of databases managed by a single server (connection pooling).

The **IOThreads** configuration item defines the maximum number of concurrent i/o operations. The default is 4. This number may need to be increased depending on database size or number of concurrent users.

The **BufferCache** configuration item specifies the cache size in megabytes. The cache is used to reduce the number of disc accesses. The default (and min.) cache size is 5 MB.

The **CheckPtSize** configuration item specifies the amount of disk space used for the transaction journal. The default size is 10 MB. When this space is exhausted the server performs an internal checkpoint operation.

Create the Volume Files

Required

```
# dbvolcreate /var/opt/eloquence6/data01.vol  
# dbvolextend -t log /var/opt/eloquence6/log.vol
```

Optional

```
# dbvolextend -t data /var/opt/eloquence6/data02.vol  
# dbvolextend -t data /var/opt/eloquence6/data03.vol  
...
```

The next step is to create the volume files. The Eloquence data base does not reside in the file system but uses *volume files* which reside in the file system as a container for the data and structural information. Please note that you need both a data volume and a log volume in order to start the data base server.

These commands should be executed by the system administrator (root) or the eloqdb user.

The dbvolcreate utility is used to create the primary data volume and initialize the server catalog.

The dbvolextend utility is used to add the transaction log volume and additional data volumes as necessary. A single volume file is currently limited to 2 GB on HP-UX and Linux and additional data volumes may be required. If your data exceeds 2 GB in size, additional data volumes must be created.

The volume filename and paths shown on the slide are arbitrary and different ones can be used.

The volume files are created and then added to the server configuration file to the [volumes] section by the dbvolcreate and dbvolextend utilities.

Start the Database Server (HP-UX)

Start the default database server

```
# /sbin/init.d/eloq6 start
```

Check if the server is active

```
# ps -ef | fgrep eloqdb6
```

Stop the default database server

```
# /sbin/init.d/eloq6 stop
```

© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page
178

After creating the database environment, the server can be started.

On HP-UX, the database server is started by executing

- `/sbin/init.d/eloq6 start`

You must have the “superuser” (root) capability to run the program.

To stop the database server, the stop argument is used.

You can verify if the server is active by issuing the command

- `ps -ef | fgrep eloqdb6`.

You should see

```
eloqdb 27987      1 0 13:04:20 ?    0:00 eloqdb6: active
eloqdb 27989 27987 0 13:04:21 ?    0:00 eloqdb6: io thread 27987:1
eloqdb 27990 27987 0 13:04:21 ?    0:00 eloqdb6: io thread 27987:2
eloqdb 27991 27987 0 13:04:21 ?    0:00 eloqdb6: io thread 27987:3
eloqdb 27992 27987 0 13:04:21 ?    0:00 eloqdb6: io thread 27987:4
```

If the database server did not start, check the syslog for any error messages.

Troubleshooting

The database server writes diagnostic messages to the syslog

- HP-UX: `/var/adm/syslog/syslog.log`
- Linux: `/var/log/messages`
- Windows: application event log

The LogFlags configuration item defines the log level

- increase the log level to informational messages by setting `LogFlags = *1` in the server configuration file.

By default, the Eloquence database server writes diagnostic messages to the syslog or a log file specified in the server configuration file. In case a problem is detected, messages in the log file provide additional information.

To increase the log level, the LogFlags configuration item can be. The star 1 logs information messages in addition. By default only critical messages are written to the log.

installation on
Linux

- Installing the product
- Kernel Parameter
- Configuring automatic server startup
- Start the database server

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 120

This section covers the installation on the Linux operating system. In addition, we mention Linux kernel parameter, show how to configure the automatic server startup and how to start the database server.

Linux Installation

Eloquence uses the RPM package manager for installation

- RedHat Linux 7.3 and SuSE Linux 8.0 have been certified with the A.07.00 release
- Other Linux distributions can be used but additional manual steps may be required

Eloquence uses the RPM package manager for installation.

Eloquence A.07.00 has been tested with RedHat Linux 7.3 and SuSE Linux 8.0. These are the most common choices for Linux server installation in the US and Europe at this time.

Other Linux distributions can be used (and are supported), such as Debian, however additional manual configuration steps may be required. This includes installing the scripts to start the database server.

Linux Installation

For installation or update execute the command below

```
$ rpm -U B1368B-A0700.glibc2.1-1.i386.rpm
```

- Temporary license option is not available during installation

On the Linux platform you should install the product archive which is compatible with your libc version (that's the standard C library) on your system. Recent Linux distributions use the glibc2.2 library.

For a recent Linux distributions you should use the glibc2.1 build. If a glibc2.2 based build of Eloquence becomes available this should be used preferably.

A separate glibc2.0 based build is available for older Linux distributions.

Please note that the option to create a temporary license key during installation is not available on Linux.

Linux Kernel Parameter

Linux uses similar kernel parameters as HP-UX

The default Linux kernel parameters are often sufficient for small to medium installations

Use `ipcs -l` to list current shared memory/semaphore limits

Kernel parameters can be changed by writing the new value to appropriate file in the `/proc/sys/kernel/` directory

On Linux similar kernel parameters are used as on HP-UX. However, the default Linux kernel parameters are often sufficient for small to medium installations.

Use the `ipcs -l` command to list current shared memory/semaphore limits.

Recent versions of the Linux kernel allow to change kernel parameters by writing the new value to the appropriate file in the `/proc/sys/kernel` directory.

The example shows how to define new semaphore related kernel parameters.

```
# echo "250 256000 32 1024" > /proc/sys/kernel/shm
```

The following kernel parameters may need to be adapted on Linux.

<code>shmmni</code>	- max. number of shared memory segments
<code>shmmax</code>	- max. size of a shared memory segment
<code>shmall</code>	- max. total shared memory (pages)
<code>sem</code>	- semaphore related parameters:
- <code>semmsl</code>	- max semaphores per array
- <code>semmns</code>	- max semaphores system wide
- <code>smopm</code>	- max ops per semop call
- <code>semmni</code>	- max number of arrays

The procedure to enable new values at boot time depends on the Linux distribution.

Linux Configuration Automatic server start

Modify the startup configuration file
/etc/sysconfig/eloquence6

```
START_ELOQSD=0  
START_ELOQDB6=1
```

To automatically start the database server, set the `START_ELOQDB6` variable to one. The other variables should be set to zero.

The configuration settings shown here enable the automatic start of the Eloquence database and disables other Eloquence services which which are not related to the database.

Linux

Start the database server

Start the default database server

- `/etc/init.d/eloq6 start`

Check if the server is active

- `/etc/init.d/eloq6 status`
- `ps ax | fgrep eloqdb6`

Stop the default database server

- `/etc/init.d/eloq6 stop`

To manually start the database server on Linux, execute the `/etc/init.d/eloq6 start` command.

To verify the Eloquence database server is active, you can either use the `ps` utility or the `/etc/init.d/eloq6 status` command.

To stop the database server, execute the `/etc/init.d/eloq6 stop` command.

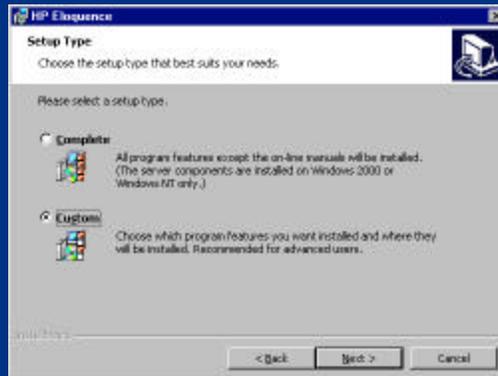
Installation on Windows

- Installing the product
- Configuring automatic startup for the database service
- Starting the database service

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 186

This section covers the installation on the Windows operating system. In addition, we show how to configure the automatic server startup and how to start the database server.

Windows Installation

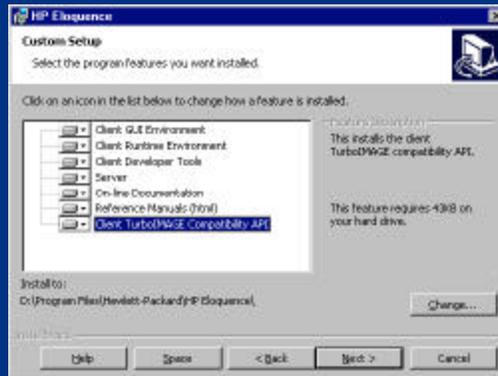


Eloquence uses the Microsoft Installer on the Windows platform. So the installation procedure should be familiar to Windows users.

During installation you have the option to install the default components or select specific components in the custom installation dialog. The setup program also has the option to modify or remove an existing installation.

A separate setup program is used for different Windows versions. The Microsoft Installer engine is included with Windows 2000 and above but must be installed on Windows NT before proceeding with the installation. This is taken care of by the setup program.

Windows Installation Custom installation



© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page
188

The custom setup allows to selectively install the required components or features.

A minimum installation requires the client runtime and TurboIMAGE components. The client developer tools component contains the database utilities.. The Server component includes the database server.

Minimal installation

- Client runtime environment
- Client TurboIMAGE compatibility API

Database utilities

- Client Developer Tools

Database server

- Server

Windows Configuration

The installation program has registered the “HP EloqDB6” service

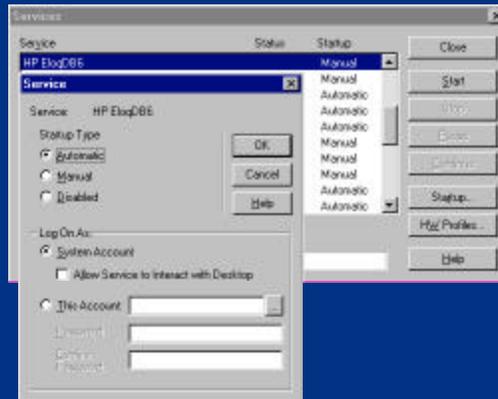
Configure automatic start mode for the Eloquence database in the service control panel

Start the service manually for the first time

During installation the Eloquence database is registered as a service. To configure automatic server startup, please open the service control panel and configure the service to start automatically.

After creating the database environment, start the service manually. In case the service startup fails, please check the windows application event log for any messages.

Windows Configuration Automatic server start



© 2002 Marxmeier Software AG

EQ HP3K Training.ppt August 9, 2002

page
100

To start the database service open the Services applet in the Control Panel or the System Management Console on Windows 2000

Then locate the “HP ElogDB6” entry in the service list and select it.

You can configure the startup options by selecting the Startup... button:

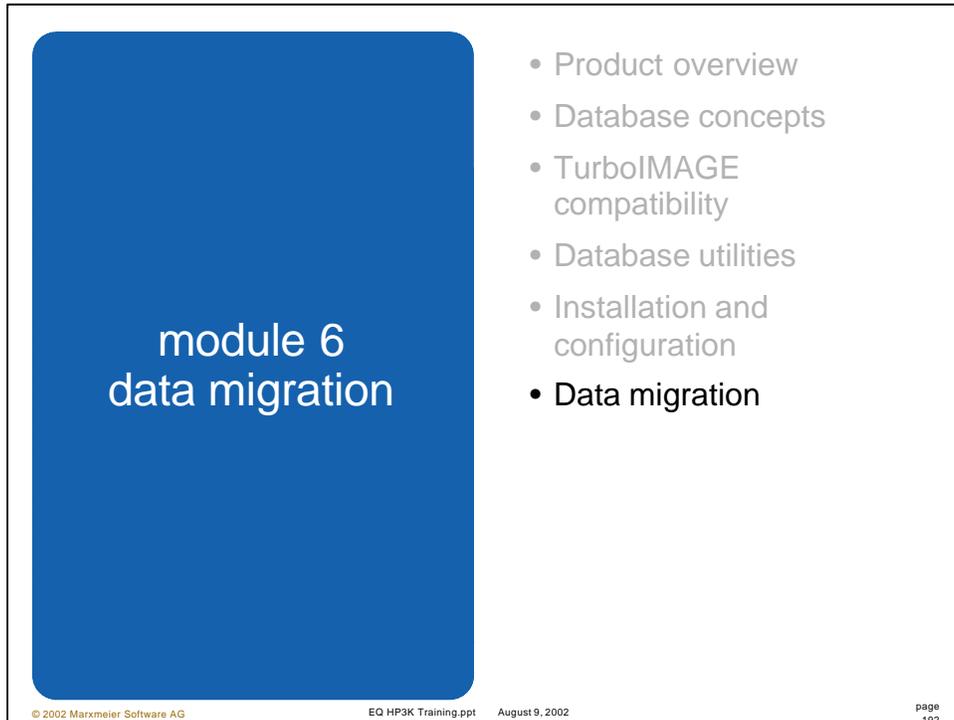
- To automatically start the database service each time Windows is restarted, select the Automatic Startup Type. This is the recommended setting.
- If the Manual Startup Type is selected the database service is not started automatically.
- By default, the “Log On As” option is set to “System Account”. Alternatively, you can specify a different account using the “This Account” option.

To run more than a single database server instance on the same system, each database environment must be registered with a unique service name. Please refer to the documentation for details.

end of module 5 installation and configuration

- Product overview
- Database concepts
- TurboIMAGE compatibility
- Database utilities
- **Installation and configuration**
- Data migration

In this module we have covered the installation of the eloquence database on the HP-UX, Linux and Windows operating system. We explained how to configure automatic server startup and how to setup the database environment.

A presentation slide with a blue rounded rectangle on the left containing the text "module 6 data migration". To the right is a bulleted list of topics. At the bottom, there is a footer with copyright information, a file name, a date, and a page number.

module 6
data migration

- Product overview
- Database concepts
- TurboIMAGE compatibility
- Database utilities
- Installation and configuration
- Data migration

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 107

This module explains how to move your database from TurboIMAGE to Eloquence.

It provides an overview on the Eloquence data migration tools for MPE, explains how to install and use them, shows how to transfer the data and how to create your database on Eloquence.

data migration

- Eloquence schema processor can use TurboIMAGE schema files
- The DBEXPORT utility is available on MPE to unload your database
- Transfer the schema and export files to Eloquence and import the data

© 2002 Marxmeier Software AG EQ HP3K Training.ppt August 9, 2002 page 101

Moving your database from TurboIMAGE to Eloquence is a straight forward process.

- The Eloquence schema processor is able to use TurboIMAGE schema files and utilities can export the database to a format used by the dbimport utility. the export file format is used to move the data.

DBEXPORT utility

- The DBEXPORT is used on MPE to export a TurboIMAGE database to text files
- The Eloquence dbimport utility makes use of the export files to load a database
- The export file format is documented and other tools can be used to create compatible files

The dbexport utility on MPE is the similar to the Eloquence dbexport utility and writes the database content in one or multiple text files.

Since the export file format is documented other ways to create the export file can be used as well.

DBEXPORT options

Usage: DBEXPORT [options] database [set ...]

options:

- help - show usage (this list)
- u user - set user identifier
- p pswd - set password
- o path - set output directory (not single file)
- v - verbose output
- c - chained export
- r - output restructure information
- s file - output into single file, '-' = stdout
- f sep - field separator, default is ','

DBEXPORT options

By default the export files are named `DBNAME.###.exp` and are created in the current directory

- The `-p passwd` option specifies the database password used to open the data base
- The `-o` option allows to specify a different target directory
- The `-s file` option specifies a single file should be used which contains all data sets
- The `-v` option displays the progress
- Any data set name(s) or number(s) specified after the database name indicates only the specified data sets should be exported

The `dbexport` utility allows to export the database content into text format.

By default a separate file is created per data set, named `DBNAME.###.exp` (where `###` is the data set number). Unless the option `-o` is specified the files are created in the current directory.

If the `-s` option is specified all output will be written to a single file or sent to another process (`'-'` specifies `stdout`).

If the `-r` option is specified additional information on item names and item types is embedded in the file.

The `dbexport` utility performs a serial `DBGET` on each data set (or the master set specified in the primary path if chained export is selected) and converts the entries into text format.

DBEXPORT utility

```
: DBEXPORT "-v -p secret TESTDB"  
Processing database : TESTDB
```

DATA SET		RECORDS	COUNT
CUSTOMERS	001 M	1177	1177
PARTS	002 M	182	182
ID	003 A	47	
ORDERS	004 D	47	47
LINEITEMS	005 D	136	136

DBINFO utility

- DBINFO lists the data sets for the specified database including data set name, type, number of entries and capacity

DBINFO options

Usage: DBINFO [options] database [set ...]

options:

```
-help   - show usage (this list)
-u user  - set user identifier
-p pswd  - set password
```

- The *-p passwd* option specifies the database password used to open the data base
- Any data set name(s) or number(s) specified after the database name indicates only the specified data sets should be displayed

DBINFO utility

```
: DBINFO "-p secret TESTDB"  
Processing database : TESTDB
```

SET NAME		RECLEN	CAPACITY	ENTRIES
CUSTOMERS	001 M	112	1355	1177
PARTS	002 M	53	524	182
ID	003 A	2	2259	47
ORDERS	004 D	21	1008	47
LINEITEMS	005 D	22	1008	136

MPE Migration tools

- The MPE DBINFO and DBEXPORT utilities are available as source code
- Compiled versions for MPE 6.0, 6.5 and MPE 7.0 are available in a tar or STORE archive format
- For download please refer to <http://www.hp-eloquence.com/download/>

Installation on MPE

To install the tar archive on your system

```
$ tar -xvf hp3kbin70.tar
```

- tar may issue a warning about being unable to restore the user and group which can be safely ignored
- This extracts the files in the current directory

To install the STORE archive on your system

```
: FILE HP3KBIN=./HP3KB70;DEV=DISC  
: restore *HP3KBIN;;LOCAL;SHOW
```

- This extracts the files in the current directory

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 and import the database

- On Eloquence, run the schema and dbcreate utilities to create the database
- Then use dbimport to load the database from the export files
- To use Eloquence indexes add them to the schema file or use the dbutil utility to add them subsequently

Eloquence defines indexes in the schema file which can be accessed with the TPI modes. To use Eloquence indexes the index definitions can either be added to the schema file or the Eloquence dbutil utility can be used to add indexes subsequently.

Create the database

Run the Eloquence schema processor

```
$ schema -T schemafile
```

- Option -T selects TurboIMAGE compatibility mode
- If you omit the -T option the schema file is understood as Eloquence schema file
 - schema may succeed but the resulting database will likely be incompatible
 - Eloquence item sizes are specified in bytes while TurboIMAGE uses halfwords, bytes or nibbles, depending on item type.

A common mistake is to omit the -T command line option with a TurboIMAGE schema file. Since Eloquence schema files are similar to TurboIMAGE schema files, the schema processor may succeed but the resulting database may be incompatible.

Create the database

After running schema, you need to create the database

```
$ dbcreate database
```

Import the data

Use dbimport to load the database

```
$ dbimport -v 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

Consider to disable SyncMode during dbimport to increase performance

end of module 6 data migration

- Product overview
- Database concepts
- TurboIMAGE compatibility
- Database utilities
- Installation and configuration
- Data migration

more information

Eloquence web site
<http://www.hp-eloquence.com>

get in contact
info@hp-eloquence.com

support
support@hp-eloquence.com

August 9, 2002

© 2002 Marxmeier Software AG EQ HP3K Training.ppt page 210

Thank you for your attention.

More information about Eloquence is available at the Eloquence web site at <http://www.hp-eloquence.com>

If have any questions, please feel free to contact the Eloquence team at info@hp-eloquence.com

