eloquence Partner Conference

Italy, June 13, 2003

Agenda

- Welcome
- Eloquence B.07.00 – Overview and features
- Lunch break
- Eloquence transition from HP to Marxmeier
- Eloquence in the future
- Discussion and presentations

Eloquence B.07.00

Overview and features

Roland Genske, Marxmeier Software AG

Overview

- Database enhancements
  - Extended database limits
  - New data types
  - Simplified tools usage
  - Enhanced QUERY program
- New start/shutdown scripts on HP-UX and Linux
- Improved dbutil program
- Forward-logging

Overview

- New platforms
  - HP-UX Itanium (IA-64)
  - Linux RedHat 8 and 9
  - Linux glibc 2.2
  - Linux Itanium (IA-64) available as beta release

Database enhancements

Extended database limits

- 2048 data items per database (was 1024)
- 500 data sets per database (was 199)
- 16 detail paths / 64 master paths per data set (was 16)
- 4096 bytes maximum item length (was 4000)
- 5120 bytes maximum entry length (was 4000)
- 8000 bytes maximum media length (was 4096)
- 256 bytes b-tree index key size (unchanged)

- Databases which use the extended limits are not backwards-compatible

New data types

- Integer data can have any length (Example: I8 = 64bit)
- K : Unsigned integer data type
- U : Upper-case string data type
  (automatically converts characters to upper-case)
- COBOL data types
  - P : Packed decimal
  - Z : Zoned decimal

New data types

- supported in Eloquence QUERY
- supported in SQL/R
- will be supported in the Eloquence language in a future release

Simplified database tools usage

Obsolete command line options –u public and –u dba

- Tools which assume –u public by default:
  - dbctl
  - dbexport
  - dbimport
  - dbinfo
  - dbtables
  - dbdumpcat
  - prschema

Simplified database tools usage

Obsolete command line options –u public and –u dba

- Tools which assume –u dba by default:
  - schema
  - dbcreate
  - dberase
  - dbpurge

Default user and password

- Default user is „public“ or „dba“
- Different default user can be specified with the EQ_DBUSER environment variable

  EQ_DBUSER=$LOGNAME

- Default password can be specified with the EQ_DBPASSWORD environment variable

  EQ_DBPASSWORD=file:$HOME/.eq_dbpassword

Default server and service

- Instead of using the –h command line option, the EQ_DBSERVER environment variable can be used

  EQ_DBSERVER=dataserv
  EQ_DBSERVER=192.168.33.1

- A service or port number can be additionally specified:

  EQ_DBSERVER=dataserv:eloqdb2
  EQ_DBSERVER=192.168.33.1:8202

Enhanced QUERY program

- A significant number of bug fixes and user contributions have been implemented
- New database limits and data types are supported
- eloqdb6 database authentication is implemented
- Internal algorithms have been optimized
- Syntax now supports mixed-case statements (upper- or lower-case)

New start/shutdown scripts on HP-UX and Linux

start/shutdown scripts

- New start/shutdown scripts on HP-UX and Linux
- Eloquence servers can be controlled separately
- Multiple eloqdb6 instances on the same machine
- Available for Eloquence B.06.32 and B.07.00

Control the Eloquence servers

- Start all Eloquence servers:
  `/sbin/init.d/eloq6 start`
- Restart the eloqsd server alone:
  `/sbin/init.d/eloq6 restart eloqsd`
- Stop the eloqdb6 database server:
  `/sbin/init.d/eloq6 stop eloqdb6`
  (This stops all eloqdb6 instances)

Multiple eloqdb6 instances

- Instance 1(„maindb“):

  First database instance
  Started at boot time
  Configuration: /etc/opt/eloquence6/eloqdb6.cfg

- Instance 2 („testdb“):

  Second test database instance
  Not started at boot time
  Configuration: /data/test/db/eloqdb6.cfg

Multiple eloqdb6 instances

- Instance 1:

```
ELOQDB6_ID[0]="maindb"
ELOQDB6_CFG[0]="eloqdb6.cfg"
ELOQDB6_START[0]=1
```

- Instance 2:

```
ELOQDB6_ID[1]="testdb"
ELOQDB6_CFG[1]="/data/test/db/eloqdb6.cfg"
ELOQDB6_START[1]=0
```

Control the eloqdb6 instances

- Stop the „maindb“ instance:
  `/sbin/init.d/eloq6 stop maindb`

- Start both the „maindb“ and „testdb“ instances:
  `/sbin/init.d/eloq6 start testdb maindb`

- Check if the „testdb“ instance is running:
  `/sbin/init.d/eloq6 status testdb`

```
eloqdb6[testdb] process is active (pid 16652)
```

Additional eloqdb6 arguments

- Common eloqdb6 arguments for all instances:

```
# Set common debug level to "*1"
ELOQDB6_DEFAULT_ARGS="-d*1"
```

- Arguments for specific instances:

```
# Use dedicated log file for "maindb" instance
ELOQDB6_ARGS[0]="–l /var/opt/eloquence6/maindb.log"

# Use dedicated log file for „testdb" instance
ELOQDB6_ARGS[1]="–l /var/opt/eloquence6/testdb.log"
```

Optional eloqdb6 prefix command

- A prefix can be defined for the eloqdb6 command line:

```
# Run the "maindb" instance at higher priority
ELOQDB6_RUNPFX[0]="/usr/bin/nice –n -10"

# Run the "testdb" instance at lower priority
ELOQDB6_RUNPFX[1]="/usr/bin/nice –n +10"
```

(Only the superuser can run processes at higher priorities, i.e. provide a negative nice value)

Show eloqdb6 instance info

- Show the configuration of the „maindb“ instance:

  /sbin/init.d/eloq6 info maindb

```
eloqdb6 instance id = maindb
    configuration file = /etc/opt/eloquence6/e...
    service = eloqdb
    args = -d*1 -l /var/opt/eloquence6/maindb.log
    run prefix = /usr/bin/nice -n –10
    automatic start = 1
```

HP-UX file location

- Script
  `/sbin/init.d/eloq6`
- Configuration
  `/etc/rc.config.d/eloquence6`
- Implementation
  `/opt/eloquence6/etc/rcfunctions`

Linux file location

- Script

  `/etc/init.d/eloq6`

  except:

  a) older SuSE Linux versions: **`/sbin/init.d/eloq6`**

  b) older RedHat Linux versions: **`/etc/rc.d/init.d/eloq6`**

- Configuration

  `/etc/sysconfig/eloquence6`

  except:

  older SuSE Linux versions: **`/etc/rc.config.d/eloquence6`**

- Implementation

  `/opt/eloquence6/etc/rcfunctions`

Important patches

- Eloquence B.07.00
  PE70-0305090

- Eloquence B.06.32
  PE63-0305150

  (Initial start/shutdown scripts did not cleanly shutdown the database servers)

Improved dbutil program

Improved dbutil program

- New interactive user interface
  - ASCII and JDLG
  - Save session to script
- Enhanced script syntax
  - Compatible with previous Eloquence versions
- On-line structural maintenance
  - Modify database schema

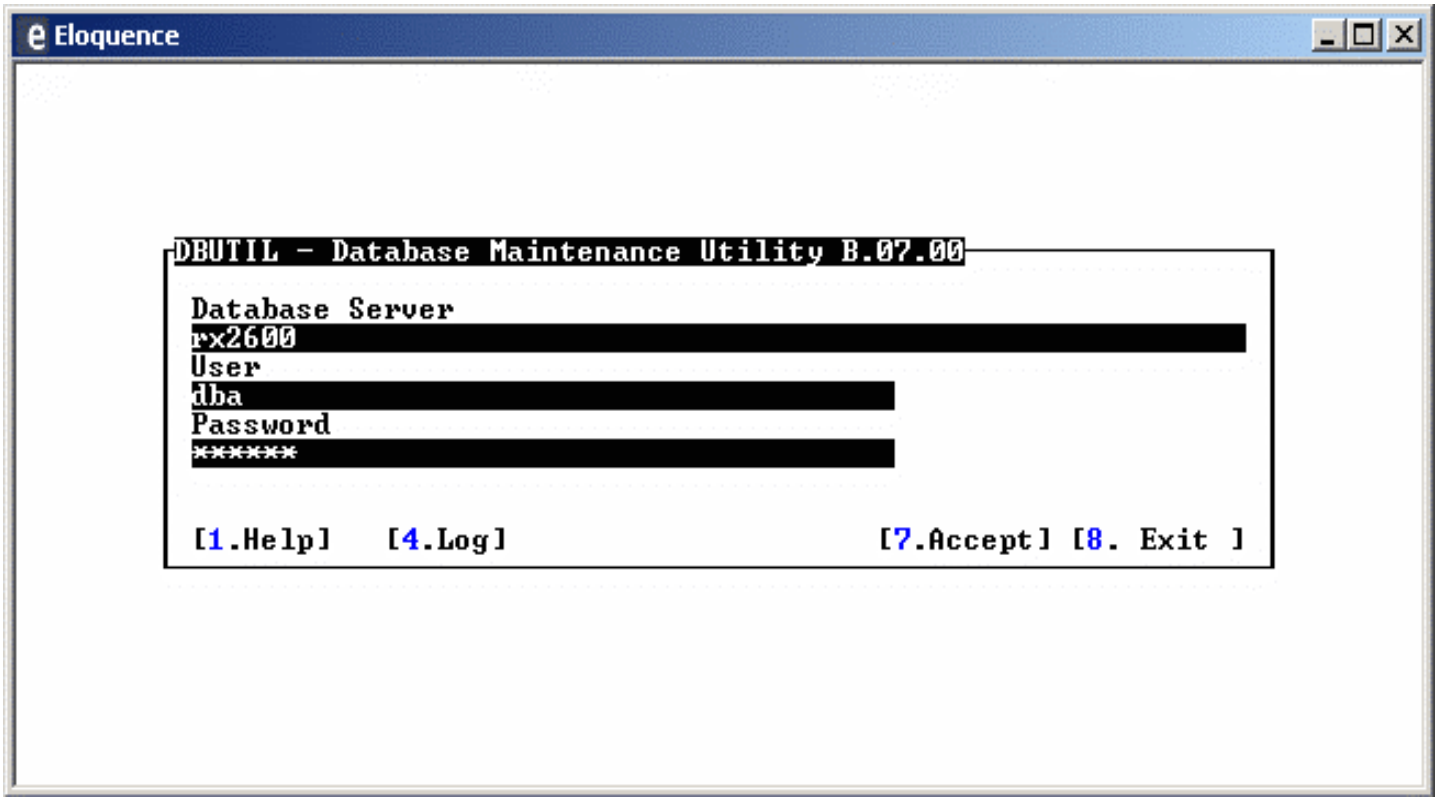
dbutil command line usage

```
usage: dbutil [options] [file|-]
   options:
   -help       - show usage (this list)
   -u name     - user name (default "dba")
   -p pswd     - password
   -h host     - hostname / ip address [:service]
   -s service  - service name or port number
   -n          - pretend (batch mode only)
   -v          - verbose (batch mode only)
   -T          - HP3000 TurboImage compatibility
```

- Optional script file or ‚-‘ (standard input)
- Interactive mode if no script file provided
- Default user is „dba“

Login dialog

Connect to the eloqdb6 server and log in:



dbutil graphical interface (JDLG)

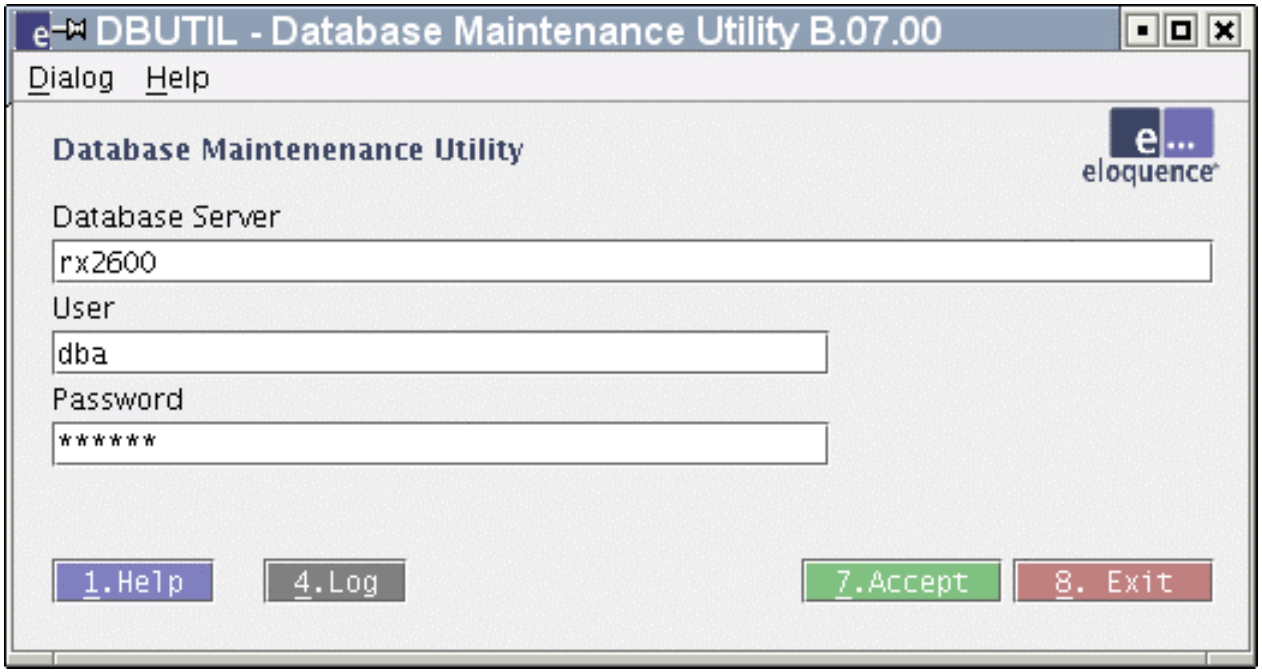
- dbutil supports JDLG to provide a GUI
- The –d command line option activates JDLG

  `dbutil –d localhost`

  (The Eloquence JDLG console must be running on the local system)

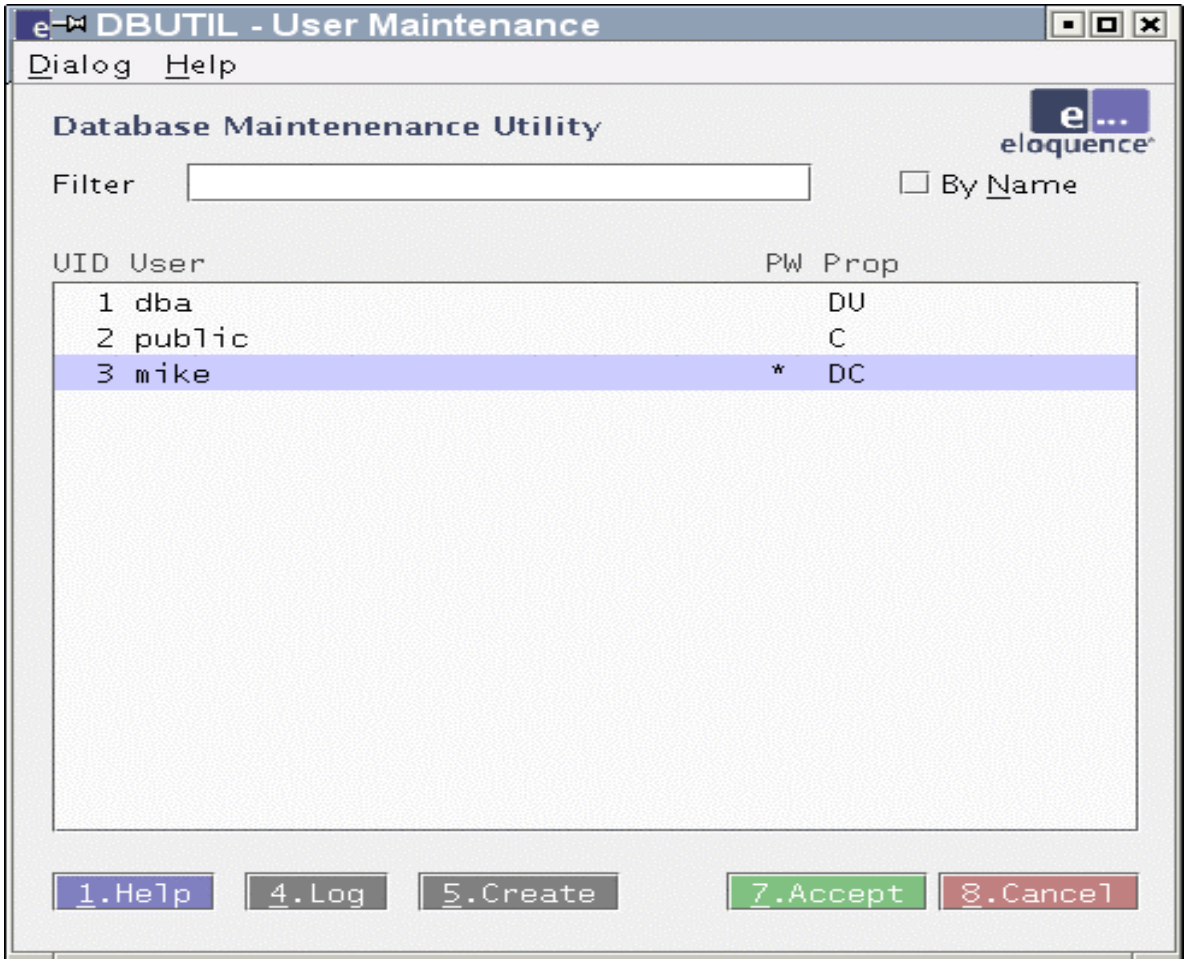
GUI login dialog

The same login dialog when JDLG is active:



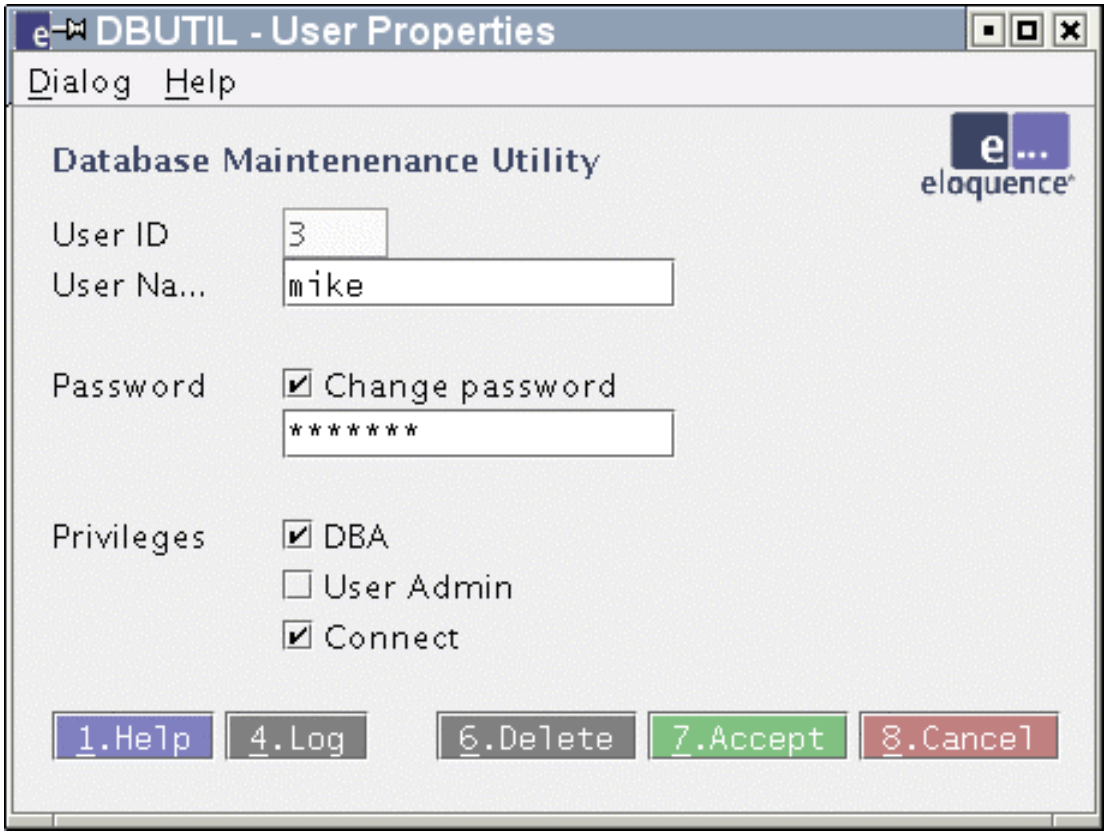
User maintenance dialog

Add new or modify existing eloqdb6 users:



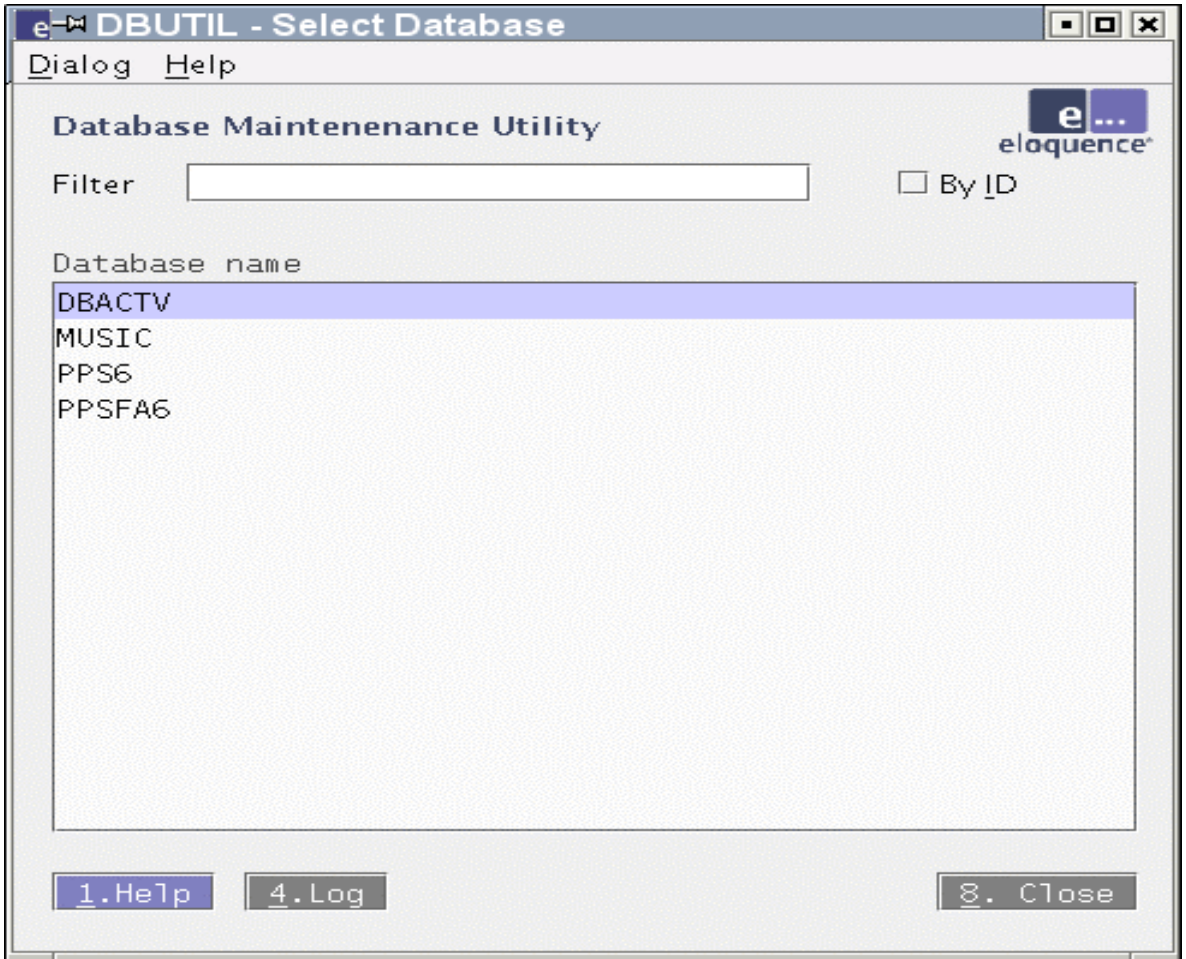
User properties dialog

Modify user privileges and assign a password:



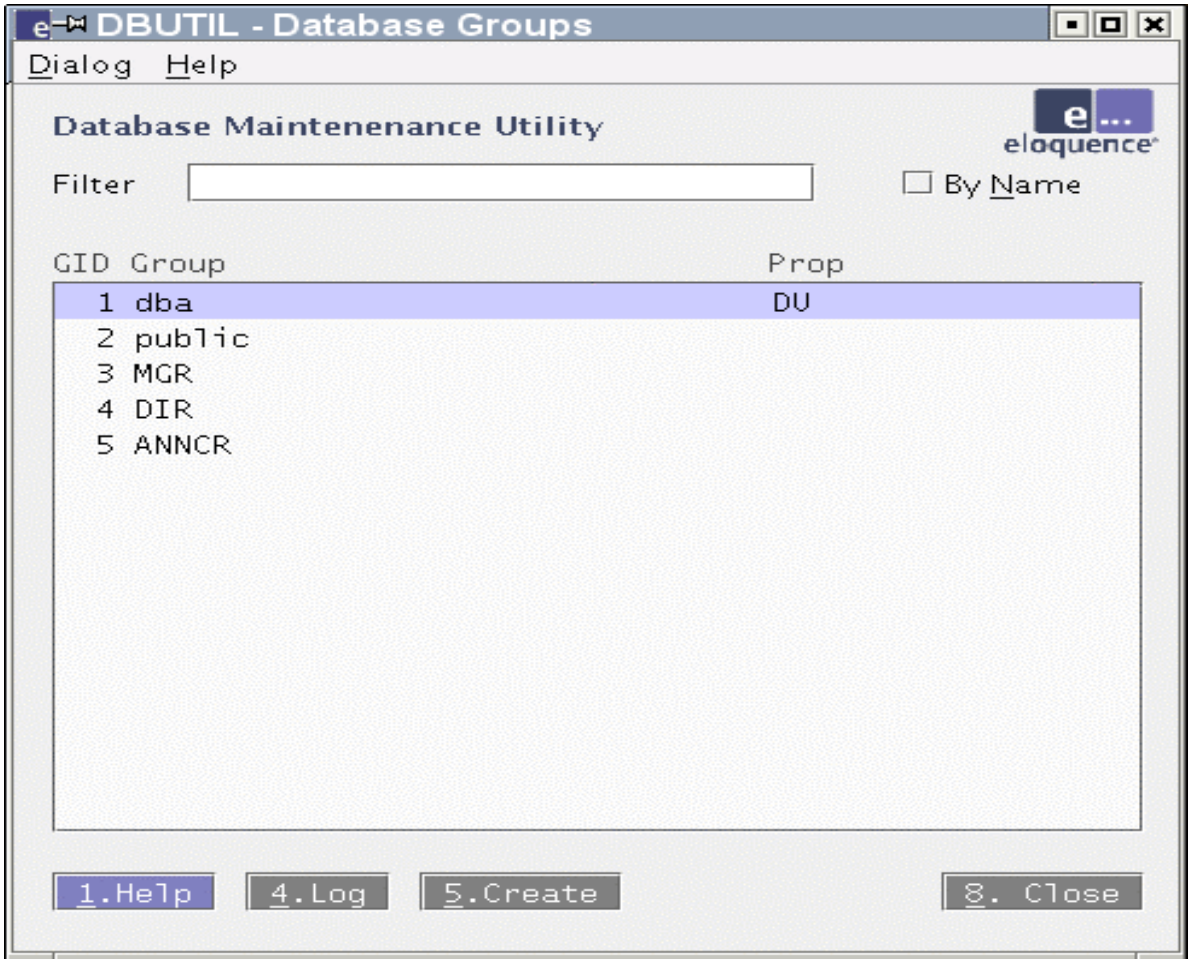
Select database dialog

Choose a database to maintain its specific settings:



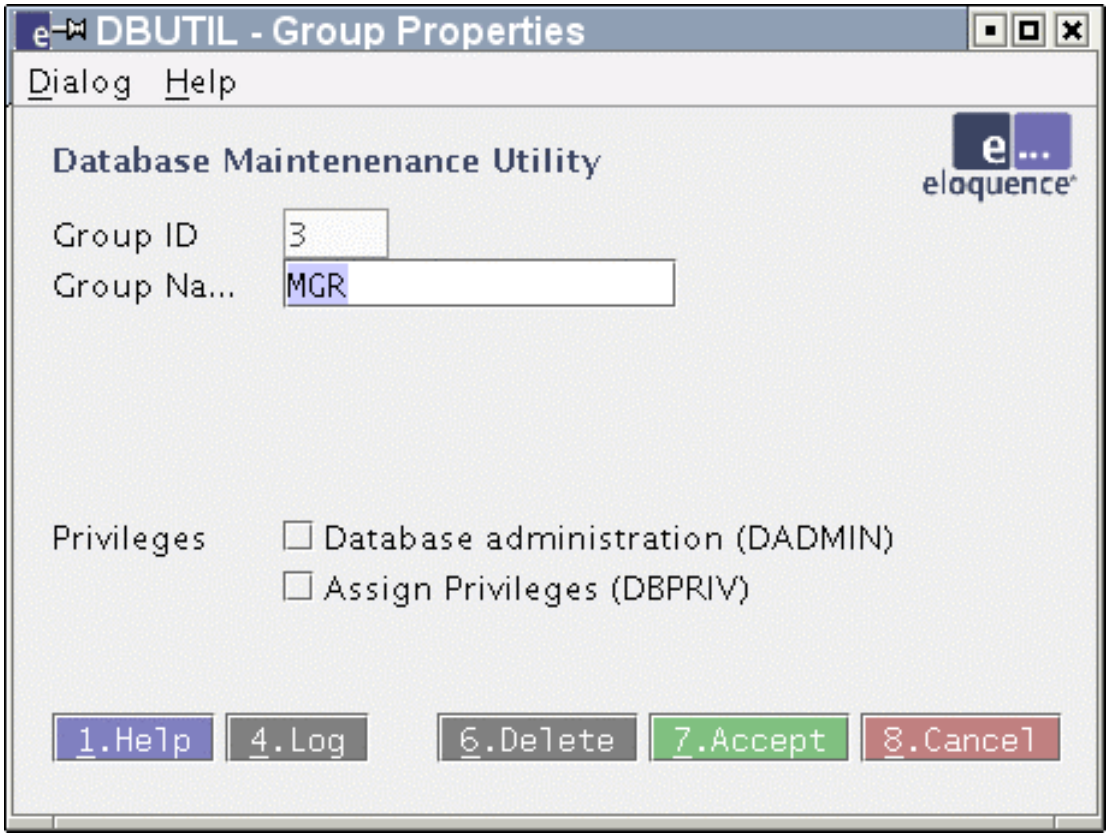
Database groups dialog

Add new or modify existing database-specific group:



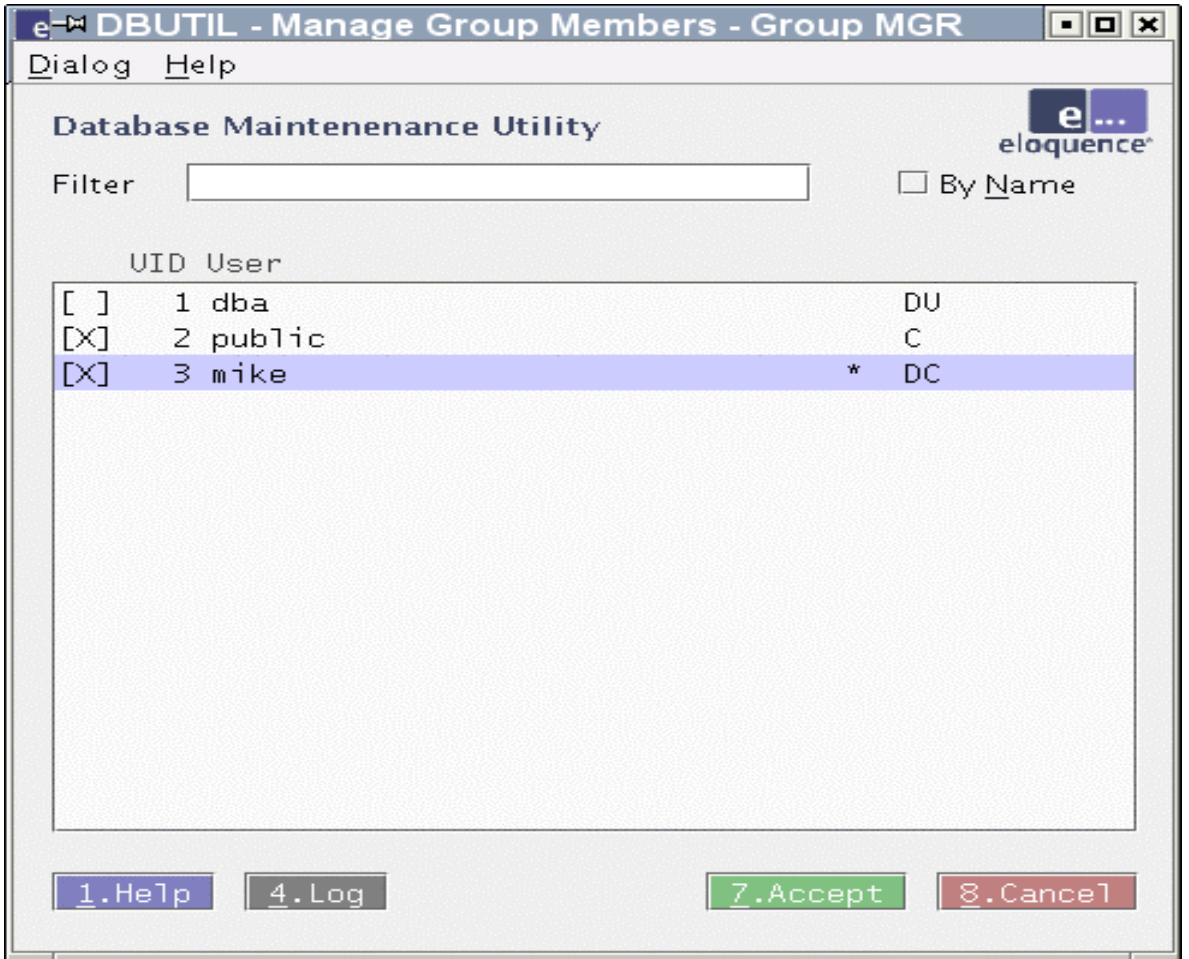
Group properties dialog

Modify group privileges:



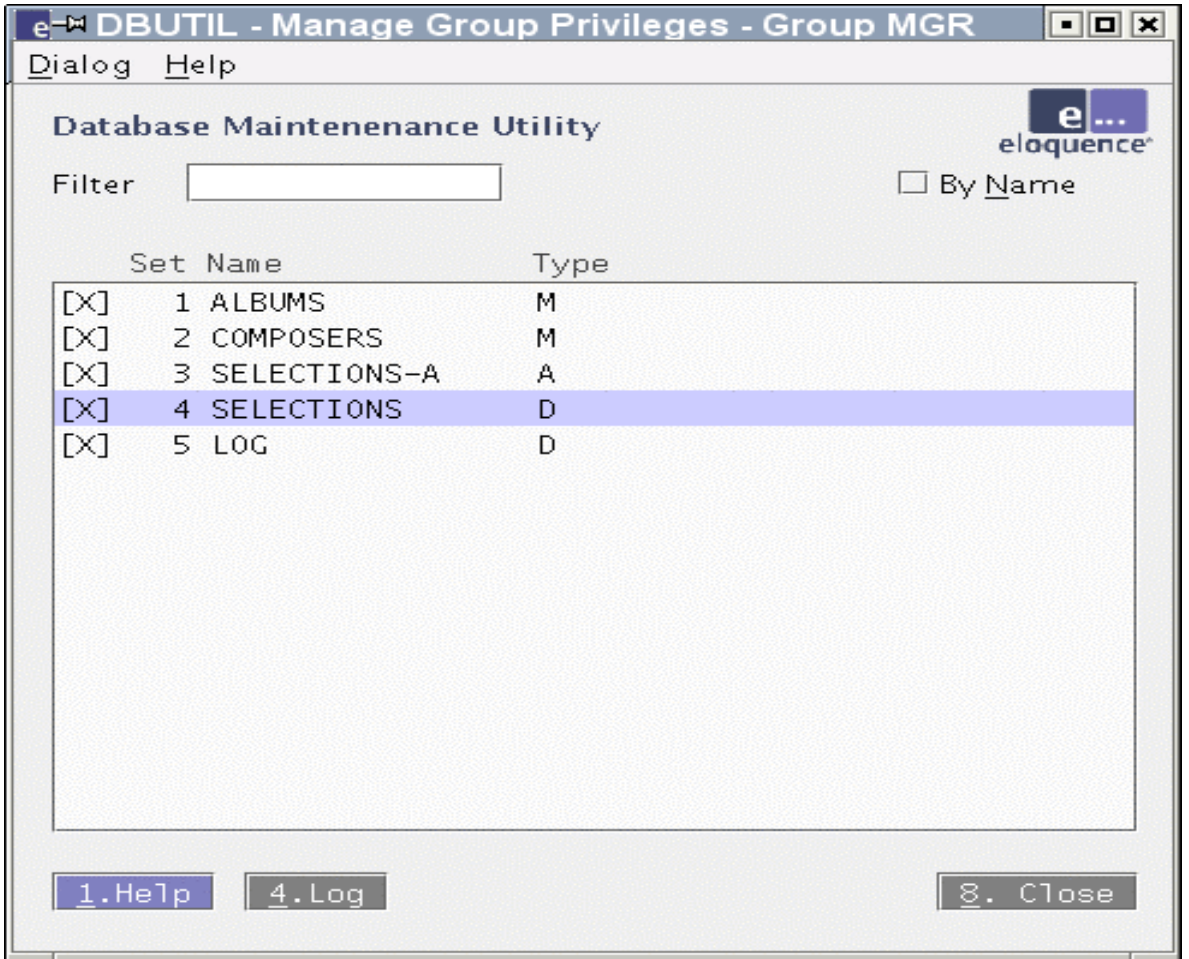
Manage group members dialog

Connect global users to database-specific groups:



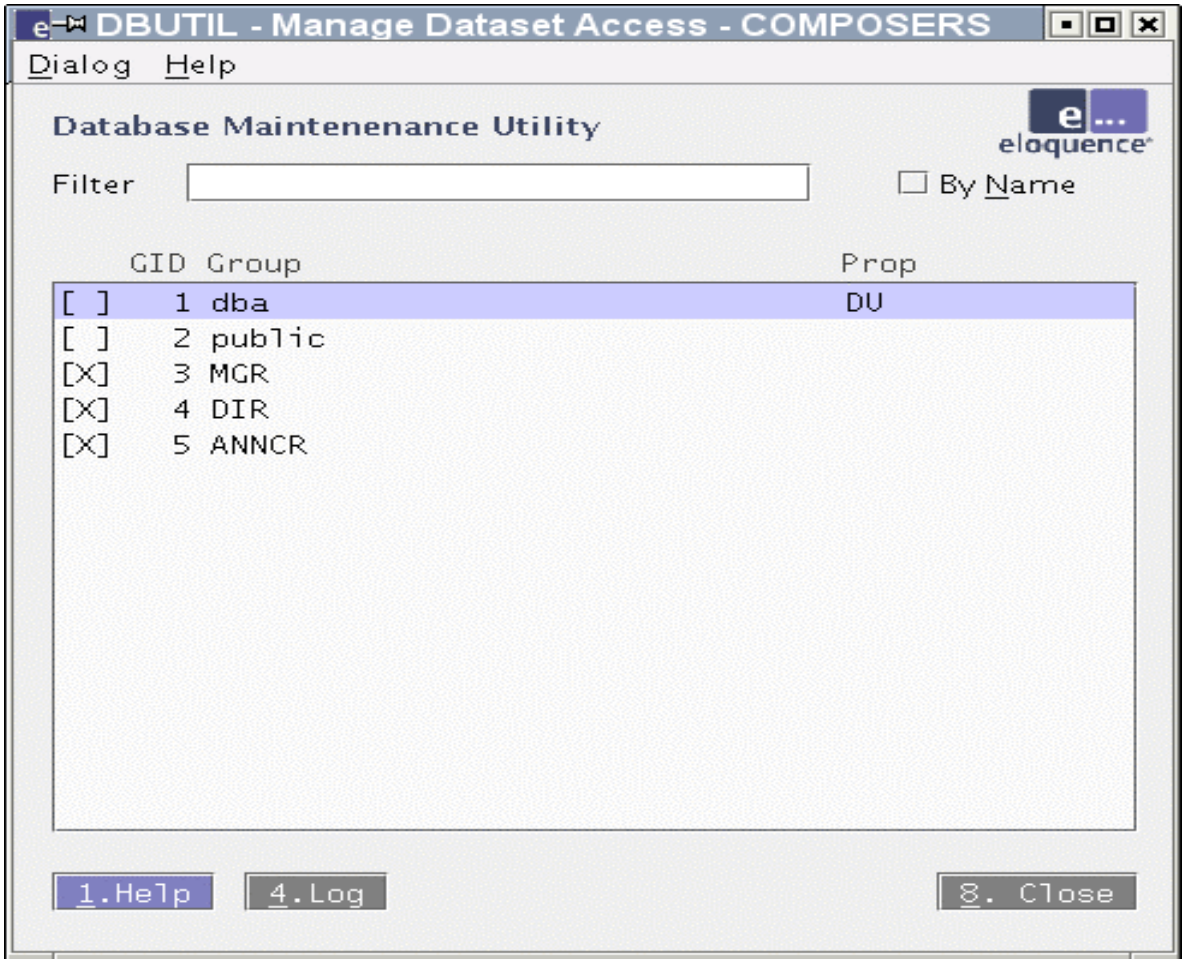
Manage group privileges dialog

Edit group-specific data set privileges:



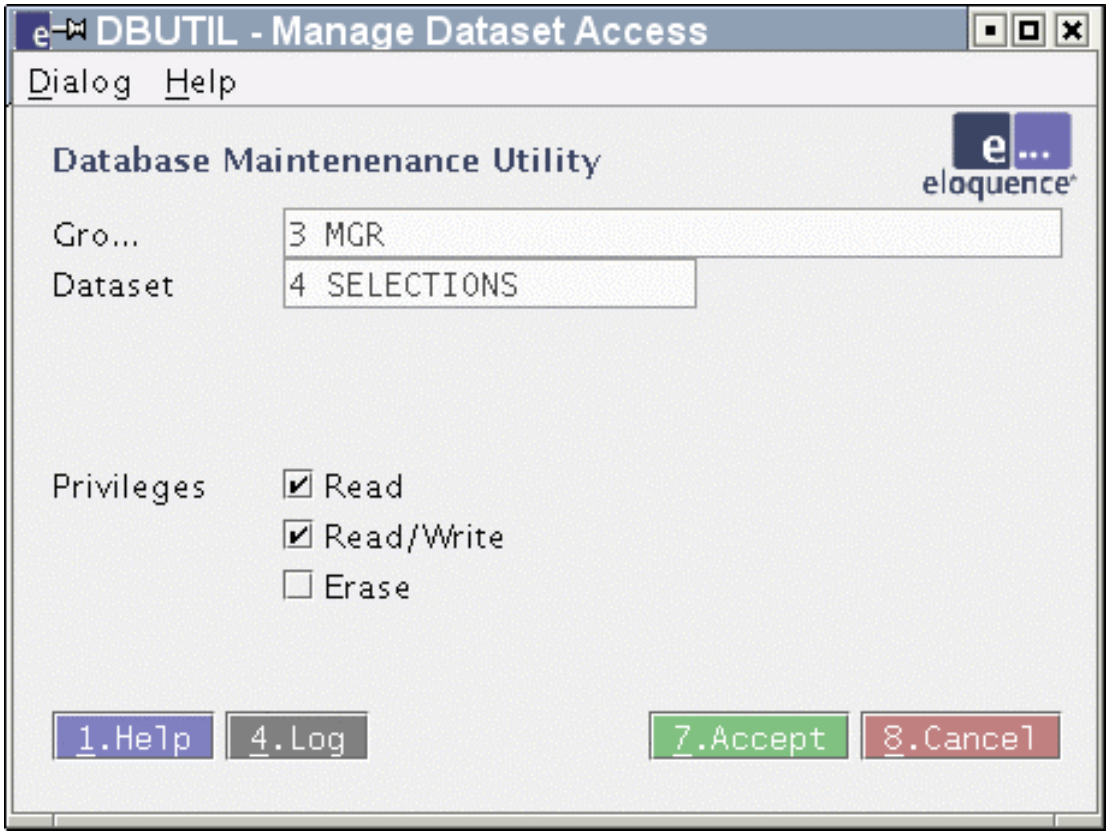
Manage data set access dialog

Or edit which group may access a specific data set:



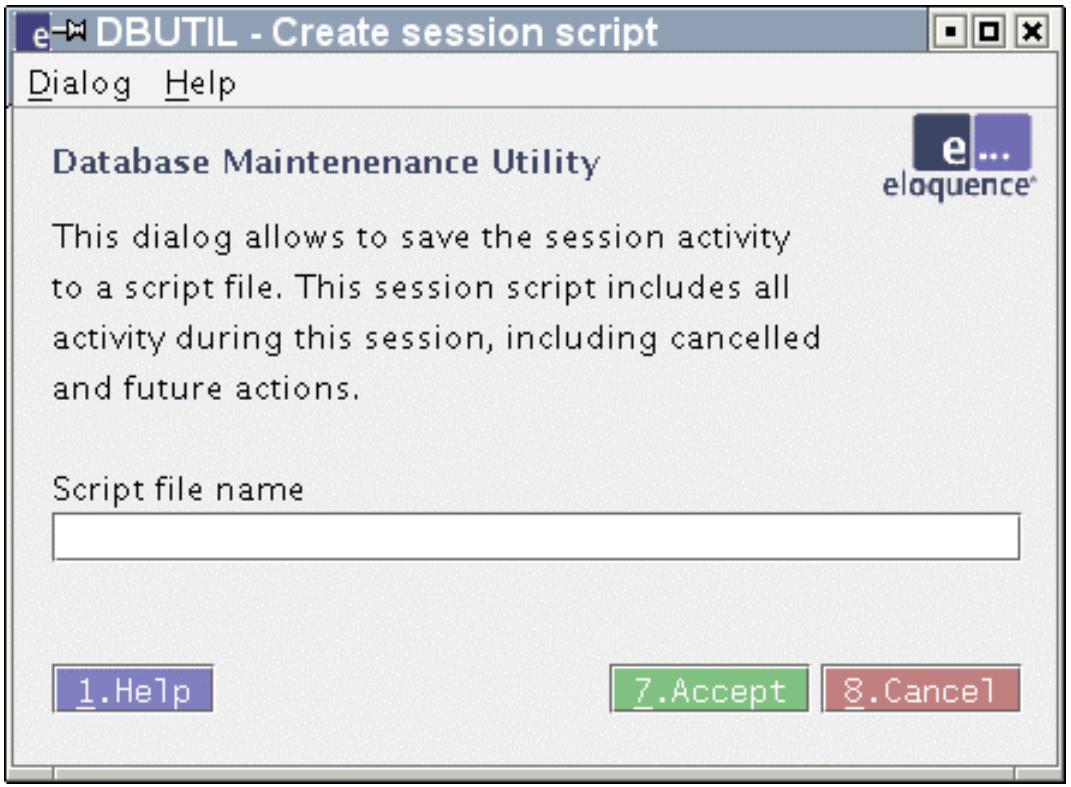
Manage data set access dialog

Modify data set privileges for a specific group:



Create session script dialog

Save interactive session to a script file:



Structural database maintenance

A database schema can be modified on-line. Examples:

- Rename a database
- Add a database-specific collating sequence
- Edit data sets, data items and index items
- Associate items and paths with a data set

Rename a database

- The RENAME DATABASE statement:

```
RENAME DATABASE "oldname" to "newname";
```

- The CHANGE DATABASE statement:

```
DATABASE "oldname";
CHANGE DATABASE NAME "newname";
```

New: database collating sequence

Sorted chains are now fully functional.
They require a database-specific collating sequence.

```
CHANGE DATABASE LANGUAGE "italian@nofold";
```

- All sorted chains must be rebuilt

Edit data sets

```
CREATE SET custmrs,MANUAL {
 ADD ITEM custno(*),matchcode,name1,name2;
 ADD INDEX imatchcode;
}

CHANGE SET custmrs {
 NAME customers;
 ADD ITEM matchcode;
 ADD INDEX imatchcode;
}

DROP SET customers,parts;
```

Edit data sets

- CHANGE SET will rebuild the data set
- If a path is affected the master will be rebuilt
- To change the order of an item delete it and subsequently insert it at the desired position:

```
CHANGE SET customers {
 DELETE ITEM matchcode;
 ADD ITEM AFTER custno matchcode;
}
```

Edit data items

```
CREATE ITEM {
 matchcode, X10;
 description, X20;
}

CHANGE ITEM oldname {
 NAME newname;
 TYPE x20;
}

DROP ITEM matchcode,description;
```

Edit data items

- Before DROP ITEM it must be deleted from data sets
- If the item data type or size is changed all related data sets will be rebuilt
- The order of items cannot be changed
- The search item of a manual master cannot be modified in a way which would lead to duplicates

Edit index items

```
CREATE IITEM {
 imatchcode = matchcode;
 imatchname = matchcode:6,name:10;
}

CHANGE IITEM imatchcode = matchcode:8;

DROP IITEM imatchcode,imatchname;
```

Edit index items

- Before DROP IITEM it must be deleted from data sets
- If the item data type or size is changed all related indexes will be rebuilt
- The order of index items cannot be changed

Associate items with data set

- Primary search item, sorted chain:

```
CHANGE SET orders
    ADD ITEM orderno(!id(orderdate));
```

- Second item in manual master is search item:

```
CREATE SET customers,MANUAL
    ADD ITEM custname,custno(*);
```

Associate items with data set

- Item at specific position:

```
CHANGE SET customers
   ADD ITEM AFTER custno
         name, matchcode;
```

- Remove item from data set:

```
CHANGE SET customers
   DELETE ITEM matchcode;
```

Assign index items to data set

- Add index item to data set:

```
CHANGE SET customers
 ADD INDEX imatchcode/"italian@nofold";
```

- Remove index item from data set:

```
CHANGE SET customers
 DELETE INDEX imatchcode;
```

Manage data set paths

```
CHANGE SET orders
   ADD PATH orderno(!id(orderdate));
```

- Exclamation mark ‚!‘ indicates the primary search item

- Additional sort item can be specified in parantheses ()
  - Chain will be ordered by this and subsequent items

```
CHANGE SET orders
   DELETE PATH orderno(id);
```

Database restructuring restrictions

- Type of a data set cannot be changed
  (for example from detail to master)
- Search item of manual master cannot be modified in a way which would lead to duplicates
- Order of data sets cannot be changed
  (new data sets are appended)
  - Exception: Master sets are inserted before the first related detail set

Database restructuring restrictions

- Order of data and index items cannot be changed (new items are appended)

- Data conversion currently cannot be configured
  - New items are initially blank
  - Floating-point values are truncated when converted to an integer data type
  - Conversion of negative values into an unsigned data type (K) currently results in the absolute value
  - Conversion warning messages appear in the syslog or server log file

Important patch

PE70-0304170

(dbutil could abort with a segmentation fault on the REVOKE ALL PRIVILEGES ON ALL FROM ... statement)

Forward-logging

Forward-logging

- Forward-logging provides additional protection against system failure
- Changes since the last backup are recorded in the forward-log
- After system failure the forward-log can be applied to the last backup with the dbrecover utility
- Recovery from forward-log is fast

Why use forward-logging

- eloqdb6 log volume holds a journal of committed transactions (roll-forward journal)
- eloqdb6 uses this journal after abnormal termination to recover during the next start
- However, if the log volume is damaged or the data volumes are lost (disk crash, operating system failure) this does not work
- Such situations required to use the last backup and lose any work since then

eloqdb6 abnormal termination

1. Disk or volume space is exhausted
2. Power was interrupted or operating system has crashed (kernel panic)
3. Disk failed, Volume files were deleted or overwritten

eloqdb6 abnormal termination

1. Disk or volume space is exhausted

- The built-in recovery will succeed when the problem is solved and the server is restarted

eloqdb6 abnormal termination

2. Power was interrupted or operating system has crashed (kernel panic)

- The log volume might be inconsistent if the problem happened at improper time (e.g. during disk write)
- In this case, the built-in recovery will not succeed
- If the eloqdb6 SyncMode is activated it is more likely that the log volume remains consistent

eloqdb6 abnormal termination

3. Disk failed, Volume files were deleted or overwritten

- The data and/or log volumes are most probably defective or inaccessible
- The built-in recovery will fail
- The data volumes must be recovered from the last backup
- All the work since the last backup will be lost

eloqdb6 abnormal termination

3. Disk failed, Volume files were deleted or overwritten

- If a forward-log is present, it can be used to recover all changes since the last backup
- No data will be lost

Physical forward-log format

- Single regular file
- Multiple regular files with automatic management
- Dedicated tape device
- Pipe (sending the forward-log data to another process)

Single regular forward-log file

```
[ForwardLog]
FwLog = /mnt/disk2/data/db-forward.log
```

- Limited to 2GB in current HP-UX an Linux versions (might not be enough to record all changes since the last backup)
- Always create regular forward-log files on a separate physical disk (different from the database disk)!

Dedicated forward-log tape device

```
[ForwardLog]
FwLog = /dev/rmt/c1t0d0BEST
```

- Tape device should be fast, otherwise eloqdb6 performance will suffer

Using forward-log pipe command

```
[ForwardLog]
FwLog = |gzip -c >/mnt/disk2/data/db-forward.log.gz
FwRecovery = |gzip -dc /mnt/disk2/data/db-forwar...
```

- Logging and recovery require different command lines
- The example above uses gzip to compress the forward-log on the fly

Automatic file management

- divides a forward-log into separate files which are easier to manage
- overcomes the 2GB file size limit on HP-UX and Linux
- maximum forward-log file size can be configured

Automatic file management

```
[ForwardLog]
FwLog = /mnt/disk2/data/db-forward-%N.log
```

- At runtime, the %N token is replaced with the volume generation followed by a sequence number
  - Example: **`db-forward-2746-3.log`**

- The volume generation is incremented
  - at the beginning of an on-line backup
  - when the eloqdb6 server is started

Automatic forward-log file creation

- New forward-log file is created
  - at the beginning of an on-line backup
    (because the volume generation is incremented)
  - when the eloqdb6 server is started
    (because the volume generation is incremented)

Automatic forward-log file creation

- New forward-log file is created
  - when forward-logging is restarted

    `dbctl –u dba forwardlog restart`

    (the volume generation is incremented)
  - when the configured maximum file size is reached (default is 2 GB)

Maximum forward-log file size

```
[ForwardLog]
FwLog = /mnt/disk2/data/db-forward-%N.log
FwMaxSize = 100
```

- When the current file would grow beyond 100MB a new file is created
  - Current file: **`db-forward-2746-3.log`**
  - New file: **`db-forward-2746-4.log`**

- Default maximum file size is 2GB

Forward-log error handling

In case the forward-log cannot be written
(disk is full, tape media failed, broken pipe)

1. either disable forward-logging but continue normal eloqdb6 server operation

2. or immediately stop the eloqdb6 to signal an emergency situation

In both cases, an error message is written to the syslog

Forward-log error handling

1. Disable forward-logging, continue normal operation:

```
[ForwardLog]
...
FwOnFailure = disable
```

   – Forward-logging must be manually re-enabled
     `dbctl –u dba forwardlog enable`

   – Creation of new forward-log is delayed until next on-line or off-line backup

Forward-log error handling

2. Immediately stop the eloqdb6 server:

```
[ForwardLog]
...
FwOnFailure = panic
```

   – eloqdb6 will refuse to start until the underlying problem is solved

Control forward-log operation

```
dbctl –u dba forwardlog disable
```

- Temporarily disable forward-logging, for example:
  - during dbimport
  - during dbrestore

Control forward-log operation

`dbctl –u dba forwardlog enable`

- New forward-log is not created immediately, creation is delayed until next on-line or off-line backup
- The volume generation is incremented

Control forward-log operation

`dbctl –u dba forwardlog restart`

- Immediately create a new forward-log
- The volume generation is incremented

Control forward-log operation

`dbctl –u dba forwardlog status`

- Query whether forward-logging is enabled or disabled
- Query the name of the current forward-log

Recover from forward-log

1. Copy all data volumes from last backup
2. All forward-log files since last backup must be present
3. Re-create the log volume(s)

   `dbvolextend -t log /path/to/log.vol`

4. Run the dbrecover utility

The dbrecover utility

```
dbrecover [options]
options:
  -t tmpdir   - directory used for temporary files
  -v          - verbose
  -d flags    - debug flags
  -c cfg      - configuration file name
```

- During recovery, transactions are combined into temporary files in the current directory
- The –t option allows to specify a different directory for these temporary files

The dbrecover utility

dbrecover reports the time of the last transaction:

```
ELOQUENCE DBRECOVER (C) Copyright 2002-2003
17219 actions have been successfully recovered.
Database environment is now up-to-date until
Tue May  6 12:10:08 2003.
```

The eloqdb6 server can now be started

Important patches

- eloqdb6 database server:

  PE70-0305270

- dbrecover utility:

  PE70-0305091

  (Various forward-logging handling problems and an internal recovery error have been fixed)

Thank you

Lunch break