

Recommended practices and procedures

- Avoid unnecessary downtime
- Have the information to quickly resolve issues
- Recommendations based on practical experience with complex sites
- Monitor server activity and performance
- House keeping

- Prefer a current product version for new installations
- Schedule periodic version upgrades

Why:

- New releases are typically “better”
 - Improved functionality and performance
 - Fixed issues
- Eloquence releases are supported for 5 years after initial release
- Avoid being “frozen” with an obsolete release

Install patches



- Install most recent patch bundle and recommended patches
- Install recommended HP-UX patches for new installations

Why:

- Proactive fixing of known issues and limitations
- Patches are binary compatible and should not change behavior

How:

- Download patch bundle from Eloquence web site and install
- Download superseding recommended patches and install on top

Check license key



- Make sure you have a valid license key
- Remove the PE license key from the license file
- Monitor expiration dates of temporary license keys

Why:

- Expired license key would prohibit server to come up
- PE license might limit volume size unexpectedly

How:

- Use the Eloquence `/opt/eloquence/8.x/etc/chklic` utility to verify the license file

Eloquence autostart?



- Consider if Eloquence should start by default

Why:

- ServiceGuard package may start db server itself
- Only start db server process after system is ready for use

How:

- Change startup config file `/etc/rc.config.d/eloquence8`
- Set `ELOQDB_START[x]=0`

Don't start eloqsd daemon



- Make sure eloqsd daemon is not started (unless you need it)

Why:

- eloqsd is only used when using the Eloquence language
- Don't run software (as root) that you don't need

How:

- Change startup config file `/etc/rc.config.d/eloquence8`
- Set `START_ELOQSD=0`

Eloquence start/stop limited to root?



- Consider if Eloquence start/stop should be allowed to database “owner”

Why:

- By default starting/stopping of the Eloquence database requires root privileges
- If enabled, the database owner account may be used to start/stop the database process and does not need root privileges

How:

- Change startup config file `/etc/rc.config.d/eloquence8`
- Set `START_STOP_AS_ROOT=0`

Use Eloquence start/stop scripts



- Use the Eloquence start/stop scripts to start or stop the database

Why:

- Eloquence provides start/stop scripts supporting multiple instances
- Supports checking status of all instances (eloq8x status)
- eloq8x stop waits until eloqdb stops; dbctl shutdown does not
- No need to reinvent them

HP-UX memory windows



- Consider using HP-UX memory windows when using multiple instances and the 32 bit eloqdb server process
- Memory windows are not needed when using the 64 bit server process

Why:

- HP-UX has a limitation in global 32 bit address space
- Multiple database instances could deplete the address space

How:

- Configure HP-UX kernel: `max_mem_window`
- Use `setmemwindow` in startup config `/etc/rc.config.d/eloquence8`
- `ELOQDB_RUNPFX[n]="setmemwindow -n -i10"`

HP-UX kernel parameter



- Make sure required HP-UX patches are installed
- Adjust HP-UX kernel parameter for “expected use”
 - nproc, npty, semmni, semmns, semmnu, semume, maxfiles_lim, nkthread, max_thread_proc, filecache_max / max_dbc_pct, max_mem_window

Why:

- Some kernel parameter require a reboot to change
- Avoid running into kernel limits during peak workloads

How:

- SAM or kctune

- Make sure the HP-UX buffer cache size is set reasonably

Why:

- Eloquence also benefits from OS file cache
- Make good use of unused memory in the system
- OS cache can shrink dynamically when memory is needed elsewhere
- Make sure the system has some memory left

How:

- Change the `dbc_max_pct / filecache_max` kernel parameter
- Default is 50% which IS a reasonable start value

config: enable HTTP status



- Enable HTTP status

Why:

- Obtain status information on active server

How:

- Change server config file: `/etc/opt/eloquence/8.x/eloqdb.cfg`
- Set `ServiceHTTP=<port#>`

config: set panic=exit



- Set panic = exit

Why:

- Avoid “silent” server restart in case of a problem
- Hides potential issues

How:

- Change server config file: `/etc/opt/eloquence/8.x/eloqdb.cfg`
- Set panic=exit

config: leave syncmode=1



- Leave syncmode = 1 unless you know what you're doing

Why:

- Improved resilience in case of system issues
- Extra protection with limited overhead
- Setting syncmode=0 will likely result in data corruption on system failure
- May use dbctl syncmode off/on temporarily in “special cases”

config: Use a log file per server



- Set logfile to a separate file per db server instance
- Set logflags = *1E2

Why:

- Have sufficiently detailed log information when needed
- The *1 logs most relevant informational messages
- The E2 includes application information

How:

- Change server config file: /etc/opt/eloquence/8.x/eloqdb.cfg
- Set LogFile = /path/to/logfile
- Set LogFlags = *1E2

config: collect performance data



- Consider enabling server statfile and session statfile

Why:

- Have sufficiently detailed performance information when needed
- Investigate performance and utilization subsequently

How:

- Change server config file: `/etc/opt/eloquence/8.x/eloqdb.cfg`
- Set `StatFile = /path/to/server.stats`
- Set `StatFileFlags = sa`
- Set `SessionStatFile = /path/to/session.stats`
- Set `SessionStatMode = 1`

config: BufferCache



- Start with a reasonable Eloquence BufferCache (256-512 MB)
- Specifies the server instance exclusive cache memory
- Eloquence also benefits from OS file cache

Why:

- Improper buffer cache is either wasting memory or limiting performance

How:

- Change server config file: `/etc/opt/eloquence/8.x/eloqdb.cfg`
- Set `BufferCache = 512`

config: VolumeSizeLimit



- Consider VolumeSizeLimit larger than the default 2 GB

Why:

- A smaller number of volume files may be more manageable
- Hard limit is 128 GB / volume file
- Up to 255 volume files are supported

How:

- Change server config file: `/etc/opt/eloquence/8.x/eloqdb.cfg`
- Set `VolumeSizeLimit = 8000`
- Verify that target file system and backup tools support large files

config: dbstore/dbrestore target



- Create a dbstore/dbrestore target

Why:

- Allows to use dbstore/dbrestore w/o restarting the server

How:

- Change server config file: `/etc/opt/eloquence/8.x/eloqdb.cfg`
- In section `[Devices]` `Backup = /path/to/directory`
- Create target even if pointing to non-existing directory or symlink

config: Use forward logs



- Use forward logs
- Consider enabling auditing

Why:

- Additional recovery options
- Critical to recover from last backup w/o losing data
- Allows analyzing user / application activity

How:

- Change server config file: `/etc/opt/eloquence/8.x/eloqdb.cfg`
- In section `[ForwardLog]` `FwLog = /path/to/fw-%N.log` and `EnableAudit=1`

config: GroupReadAccess



- Consider enabling GroupReadAccess

Why:

- Allows read-only access to forward logs to selected users that would benefit from access to live audit information

How:

- Change server config file: /etc/opt/eloquence/8.x/eloqdb.cfg
- In section [ForwardLog] GroupReadAccess = 1
- Assign (supplemental) group to relevant users

Create log volume(s) on different fs



- Consider creating the Eloquence transaction log volume on a different file system
- Use a different naming convention for log volume file

Why:

- Log volume gets same write activity as all data volume files combined. It is typically “write-only”
- Improved performance
- Additional protection in case of failure
- Typically does not need to be backed up

Create fw log files on different fs



- Consider creating forward log files on a different file system

Why:

- Forward log files is written to for each db change
- It is typically “write-only”
- Improved performance
- Additional protection in case of disk failure

How:

- Change server config file: `/etc/opt/eloquence/8.x/eloqdb.cfg`
- In section `[ForwardLog]` `FwLog = /path/to/fw-%N.log`

Separate directory for db env



- Use a separate directory for each db environment
- Use a different “owner” for each db environment

Why:

- Easier to maintain
- Less admin mistakes

Secure dba account



- Consider using a password for the dba account
- Create an "emergency" dba account (and document the password)

Why:

- Restrict admin access to authorized users
- Secure against "loss" of dba password
- Separate db account per individual helps tracking changes (who did it)

How:

- Use dbutil

Disable anonymous data access



- Consider disabling the “public” user
- or remove user “public” from db access groups

Why:

- Public user by default grants anonymous access to data
- Improve database security

How:

- remove connect attribute from public user (dbutil)
- REVOKE CONNECT from “public”;
- DATABASE “db”; REVOKE “group1”, “group2” FROM “public”;

Use operator instead of dba (B.08.10)



- Consider using “operator” user instead of “dba” user in batch jobs and for admin tasks

Why:

- Additional security, role separation
- The operator privilege has most of dba capabilities besides changing permissions and database structure
- For example: backup mode, switch logfile, statfile, forward log

How:

- Create new user “batch” and grant “OPERATOR” privilege
- Add “OPERATOR” privilege to existing users

Monitor disk space



- Monitor sufficient disk space is available
- Monitor volume fill grade (`dbdumpcat -S`)
- Create “spare” volume file in advance

Why:

- Prevent unplanned server downtimes

Online backup



- Use online backup
- Do not backup log volume files
- Do backup forward log files since last backup
- Plan to test recovery media and procedures periodically

Why:

- on-line backup is safe and less intrusive than other options
- helps reduce unnecessary backup downtime
- recovery fire drill verifies media+procedures and trains staff

Multiple db server instances



- Use different “owner” for each instance
- Don’t use the default instance and port

Why:

- Isolate instances at the file system level
- Protect against accidental access
- Require explicit use of EQ_DBSERVER to specify the instance

EQ_DBUSER=file:/path/to/file



- Use EQ_DBUSER pointing to a protected file
- Do not use the `-p` command line option to specify password

Why:

- prevent password disclosure (to non-root users)
- command line is visible to other users and is logged

How:

- Change file system access to only allow access by user
- Optionally remove password files on logout

Monitor server message log



- Monitor server message log for "uncommon" messages

Why:

- Server emits advisory messages for “unusual” situations

How:

- Use grep to filter common messages
- `greplog.sh eloqdb.log | mailx -s “eloqdb messages” admin_user`

Switch server message log



- Switch server message log periodically
- Keep a copy for a few days/weeks/months

Why:

- Keep the server message log at a reasonable size
- Server message log cannot exceed 2 GB
- Have the information available when needed

Switch server message log



A script as below may be used to switch the server message log, filter “unusual” messages and compress the previous log.

```
cd /path/to/logfile
ts=$(date +"%Y%m%d%H%M%S")
mv eloqdb.log eloqdb.log-$ts
dbctl -u dba logfile $PWD/eloqdb.log
greplog.sh eloqsb.log-$ts \
| mailx -s "eloqdb messages" admin_user
gzip eloqdb.log-$ts
```

Switch server stat files



- Switch server stat files periodically
- Keep a copy for a few days/weeks/months

Why:

- Keep the statistics files at a reasonable size
- Have the information available when needed

Switch server stat files



A script as below may be used to switch the server statistics files.

```
cd /path/to/statfile
ts=$(date +"%Y%m%d%H%M%S")
mv server.stats server.stats-$ts
mv session.stats session.stats-$ts
dbctl -u dba statfile $PWD/server.stats
dbctl -u dba sessionstatfile $PWD/session.stats
gzip server.stats-$ts session.stats-$ts
```

Note: The server.stats file may be moved w/o notifying the server.

More information



Detailed information is available on the

Eloquence web site

<http://eloquence.marxmeier.com>

Get in contact

info@marxmeier.com