



Symas OpenLDAP

How-To Guides

Accesslog Database Utilization

This section describes how to track how "full" the accesslog database is and provides steps to optimize the accesslog configuration to improve performance and avoid running out of space.

These examples set the accesslog database content to be retained for 3 days and then purged every day thereafter. The database maxsize is 8589934592 bytes (8 GB) and the database is only used for delta-sync replication. These settings are typical for performance in a standard operational setting.

Example Accesslog Overlay Config

```
# Access log - used for delta-sync repl too
overlay accesslog
logdb                cn=accesslog
logops               writes
logsuccess           TRUE< o:p>
logpurge             3+00:00 01+00:00
```

Example Accesslog Database Config

```
### Accesslog Database Configuration ###
database            mdb
directory           /var/symas/openldap-data/accesslog
maxsize             8589934592
suffix              cn=accesslog
rootdn              cn=accesslog
index               entryCSN,objectClass,reqEnd,reqResult,reqStart eq

overlay             syncprov
syncprov-nopresent TRUE
syncprov-reloadhint TRUE
```

Tracking Database Capacity

There are two indicators to track how "full" the accesslog database is. Both utilized the built-in `mdb_stat` function.

1. To get the actual size used, you multiply the number of pages used by the page size value. This will give you the number of bytes in use, which you can compare to the configured `maxsize` value to track how "full" the database is.

Caution: when deletes occur the pages are still flagged as used, but they are put on the freelist. As long as the rate of change and rate of deletion are near equal, it is unlikely you will see page usage grow and it will never shrink.

```
mdb_stat -e /var/symas/openldap-data/accesslog
```

Environment Info

```
Map address: (nil)
Map size: 8589934592
Page size: 4096 <-----
Max pages: 262144
Number of pages used: 675 <-----
Last transaction ID: 4
```





Symas OpenLDAP

How-To Guides

Max readers: 126
Number of readers used: 1

$$4096 \text{ (page size)} \times 675 \text{ (used pages)} = 2764800 / 8589934592 = 3.21\% \text{ Full}$$

2. To know how many pages are actively being used, subtract the freelist pages from the pages used. Then follow instructions from step 1.

```
mdb_stat -ef /var/symas/openldap-data/accesslog
```

```
Environment Info
Map address: (nil)
Map size: 8589934592
Page size: 4096 <-----
Max pages: 262144
Number of pages used: 675 <-----
Last transaction ID: 4
Max readers: 126
Number of readers used: 1
Freelist Status
Tree depth: 1
Branch pages: 0
Leaf pages: 1
Overflow pages: 0
Entries: 2
Free pages: 86 <-----
```

$$4096 \text{ (page size)} \times 589 \text{ (675 (used pages) - 86 (free pages))} = 2412544 / 8589934592 = 2.81\% \text{ Full}$$

However, the former strategy is better when considering whether or not the DB is in danger of becoming "full", because the used pages value gives you information on your "upper" bound usage, (i.e. the most pages that have ever been in full use).

For example, let's say in a average day, 500 pages are the most that are ever used. But during a special circumstance there are an above average number of writes that occur, and the pages used jumps up to 1000. Then mdb_stat will report 1000 as the "pages used", even if 500 is the average usage. This indicates that generally on the upper end, 1000 is the most pages ever in use. The purpose is revealing what the usage is when things are more extreme, as opposed to normal operations.

Recommended delta-sync replication configuration

Configure delta-syncrepl configurations to purge the accesslog multiple times a day.

For example:

```
logpurge 03+00:00 00+04:00
```

It is problematic for some clients to have the master delay write operations in a busy environment while the accesslog is purged of a full day's worth of expired changes. Instead, break it down into 6 purges a day. This minimizes the delay during the purge phase, thus reducing the performance impact for write handling and replication.

