

## Univention Corporate Server



### Performance guide



## Table of Contents

1. Introduction .....	4
2. OpenLDAP and listener/notifier domain replication .....	5
2.1. Indices .....	5
2.2. Configuration of the database backend (BDB) .....	5
2.3. OpenLDAP ACLs .....	6
2.4. Univention Directory Listener .....	6
3. Name Service Cache Daemon (NSCD) .....	7
4. Local group cache .....	8
5. UCS management system .....	9
5.1. Disabling automatic search .....	9
5.2. Imposing a size limit for searches .....	9
6. Further services and components .....	10
6.1. Squid .....	10
6.2. Bind .....	10
6.3. Kernel .....	10
6.4. Samba .....	10
Bibliography .....	12

# Chapter 1. Introduction


In its default configuration, UCS is suitable for environments with up to 5,000 users.

This document describes configuration modifications which can increase performance in larger environments.

# Chapter 2. OpenLDAP and listener/notifier domain replication

As a core element in the operation and administration of a UCS domain, the performance of the LDAP server plays a central role in the overall performance.

## 2.1. Indices


Feedback 

Comparable with other database systems, OpenLDAP runs indices about commonly requested attributes. For indexed attributes, a search is not performed via the full database contents, but over an optimized subsection.

The preset indices are often extended with new UCS releases. As of UCS 2.3-2, it is possible to update the indices automatically. If the Univention Configuration Registry variable `ldap/index/autorebuild` is set to `yes` or `true`, the LDAP indices configured on the system are expanded automatically and the `slapindex` command is used to rebuild the indices during release upgrades.

For the indexing of large directories, it must be taken into account that `slapindex` running times can take several hours.

## 2.2. Configuration of the database backend (BDB)

Feedback 

The data of the LDAP server are stored in a database in the Berkeley database format (BDB).

The BDB configuration is performed via Univention Configuration Registry variables, which configure the `/var/lib/univention-ldap/ldap/DB_CONFIG` file.

The Univention Configuration Registry variable `ldap/database/bdb/set_cachesize` defines the cache size (default 90 MB). As a general rule, the cache should always be large enough to be able to hold all indices.

The Univention Configuration Registry variable `ldap/database/bdb/set_lg_bsize` controls the maximum size of the BDB transaction log in bytes. These files are also stored in the data directory and store requested modifications before being run. If the transaction is cancelled or the LDAP server is not exited properly, the last log files can be used to check the last transactions and restore a consistent database status.

Following most changes to `DB_CONFIG`, the `db_recover` tool must be run to adopt the new configuration for the database. This tool also performs the replay of potentially missed transactions.. To ensure problem-free OpenLDAP operation, `db_recover` is run in the `slapd` init script. In other words, when BDB-specific Univention Configuration Registry variables are changed, it suffices to restart the LDAP server.

The `db4.8_stat` tool is provided to check the BDB database. On a running LDAP server, it is possible, among other things, to request information about the maximum cache size, cache utilisation and locking.

The following modifications may prove useful in larger environments:

- *Increasing the size of the cache*


A larger cache improves performance, particularly for commonly requested objects. Statistics of the cache usage can be displayed with `db4.8_stat -m -h /var/lib/univention-ldap/ldap/`.

- *Increasing the maximum number of locks*

In the default setting, BDB allows a maximum of 1,000 locks for concurrent accesses (locks), users (lockers) or objects. An evaluation of the actually used locks can be viewed using `db4.8_stat -c -h /var/lib/univention-ldap/ldap/`.

The locks can be set with the DB\_config options `set_lk_max_lockers`, `set_lk_max_locks` and `set_lk_max_objects`.

## 2.3. OpenLDAP ACLs


Feedback 

Access to the information contained in the LDAP directory is controlled by access control lists (ACLs) on the server side. General information on the configuration of ACLs in UCS can be found in the LDAP chapter of the [ucs-manual].

The default setting of the LDAP server does not allow anonymous access to the LDAP directory. If the Univention Configuration Registry variable `ldap/acl/read/anonymous` is set to `yes`, these access restrictions do not apply and the LDAP performance is increased. Access to sensitive attributes such as user passwords remains protected via separate ACLs.

Nested groups are also supported. The Univention Configuration Registry variable `ldap/acl/nested-groups` can be used to deactivate the nested groups function for LDAP ACLs, which will result in a speed increase for directory requests.

## 2.4. Univention Directory Listener

Feedback 

In the default setting, the Univention Directory Listener performs safety checks to prevent a user name being added into a group twice. These checks can be deactivated by setting the Univention Configuration Registry variables `listener/memberuid/skip` and `listener/uniquepassword/skip` to `no`.

## Chapter 3. Name Service Cache Daemon (NSCD)

Name resolutions can be cached by the *Name Service Cache Daemon* (NSCD) in order to speed up frequently recurring requests for unchanged data. Thus, if a repeated request occurs, instead of querying the LDAP server, the data are simply drawn directly from the cache.

The size of the cache held by the NSCD is preconfigured for an environment with 5,000 users. If more users or hosts are created, the cache should be enlarged as otherwise it will not be possible to cache enough entries.

The following Univention Configuration Registry variables can be set:

- `nscd/hosts/size` should be at least the same as the number of all the computers entered in the DNS.
- `nscd/passwd/size` should be at least the same as the number of users.

To allow an efficient cache allocation, the value selected should always be a prime number, in case of doubt the next highest prime number should be selected.

A script can be downloaded from <http://download.univention.de/download/scripts/nscdCachesize.sh> which suggests corresponding values based on the objects currently included in the system.

## Chapter 4. Local group cache


In the default setting, the group cache is regenerated every time changes are made to a group. This avoids cache effects whereby group memberships only become visible for a service after the next scheduled group cache rewrite (in the default setting after 15 minutes and after 15 seconds of inactivity in the Univention Directory Listener). In larger environments with a lot of group changes, this function should be deactivated by setting the Univention Configuration Registry variable `nss/group/invalidate_cache_on_changes` to `false`. This setting takes effect immediately and does not require a restart of the Univention Directory Listener.

When the group cache file is being generated, the script verifies whether the group members are still present in the LDAP directory. If only the Univention Management Console is used for the management of the LDAP directory, this additional check is not necessary and can be disabled by setting the Univention Configuration Registry variable `nss/group/cachefile/check_member` to `false`.




# Chapter 5. UCS management system

## 5.1. Disabling automatic search

Feedback 

By default all objects are automatically searched for in the domain management modules of the Univention Management Console. This behaviour can be disabled by setting the Univention Configuration Registry variable `directory/manager/web/modules/autosearch` to 0.


## 5.2. Imposing a size limit for searches

Feedback 

The Univention Configuration Registry variable `directory/manager/web/sizelimit` is used to impose an upper limit for search results. If, e.g., this variable is set to 2000 (as is the default), searching for more than 2000 users would not be performed and instead the user is asked to refine the search.


# Chapter 6. Further services and components

## 6.1. Squid

Feedback 

If the Squid proxy service is used with NTLM authentication, the authentication is performed via the Winbind service. The communication with Winbind is performed over a queue, as standard up to five running NTLM requests can be processed in parallel. If the Winbind service answers slowly or many proxy requests are received in parallel, the top limit of the queue may be reached and the Squid user may receive an authentication error. The queue size can be configured with the Univention Configuration Registry variable `squid/ntlmauth/children`.


## 6.2. Bind

Feedback 

Bind can use two different backends for its configuration: OpenLDAP or the internal LDB database of Samba 4. The backend is configured via the Univention Configuration Registry variable `dns/backend`.

When using the Samba backend, a search is performed in the LDAP for every DNS request. With the OpenLDAP backend, a search is only performed in the directory service if the DNS data has changed. For this reason, using the OpenLDAP backend can reduce the load on a Samba 4 domain controller.

## 6.3. Kernel

Feedback 

In medium and larger environments the maximum number of open files allowed by the Linux kernel may be set too low by default. As each instance requires some unswappable memory in the Linux kernel, too many objects may lead to a resource and denial-of-service problems in multi-user environments. Because of that the number of allowed file objects is limited by default.

The maximum number of open files can be configured on a per-user or per-group basis. The default for all users can set through the following Univention Configuration Registry variables:

`security/limits/user/default/hard/nofile` The hard limit defines the upper limit a user can assign to a process. The default is 32768.

`security/limits/user/default/soft/nofile` The soft limit defines the default settings for the processes of the user. The default is 32768.


A similar problem exists with the *Inotify* sub-system of the kernel, which can be used by all users and applications to monitor changes in file systems.

`kernel/fs/inotify/max_user_instances` The upper limit of Inotify services per user ID. The default is 511.

`kernel/fs/inotify/max_user_watches` The upper limit of files per user which can be watched by the Inotify service. The default is 32767.

`kernel/fs/inotify/max_queued_events` The upper limit of queued events per Inotify instance. The default is 16384.

## 6.4. Samba

Feedback 

Samba uses its own mechanism to specify the maximum number of open files. This can be configured through the Univention Configuration Registry variable `samba/max_open_files`. The default is 32808.

If the log file `/var/log/samba/log.smbd` contains errors like `Failed to init inotify - Too many open files`, the kernel and Samba limits should be increased and the services should be restarted.

# Bibliography

[ucs-manual] Univention GmbH. 2014. *Univention Corporate Server - Manual for users and administrators*. <http://docs.univention.de/manual-3.2.html>.