univention
be open.

# Univention Corporate Server

## Manual for users and administrators

www.univention.de

Version 4.4-9
Date: April 12th, 2022

Alle Rechte vorbehalten./ All rights reserved.
(c) 2002-2022
Univention GmbH
Mary-Somerville-Straße 1
28359 Bremen
Deutschland
feedback@univention.de

Jede aufgeführte Marke und jedes Warenzeichen steht im Eigentum ihrer jeweiligen eingetragenen Rechtsinhaber. Linux ist ein eingetragenes Warenzeichen von Linus Torvalds.

The mentioned brand names and registered trademarks are owned by the respective legal owners in each case. Linux is a registered trademark of Linus Torvalds.

univention
be open.

# Table of Contents


1. Introduction ..... 13
    1.1. What is Univention Corporate Server? ..... 13
    1.2. Overview of UCS ..... 14
        1.2.1. Commissioning ..... 14
        1.2.2. Domain concept ..... 14
        1.2.3. Expandability with the Univention App Center ..... 15
        1.2.4. LDAP directory service ..... 15
        1.2.5. Domain administration ..... 16
        1.2.6. Computer administration ..... 17
        1.2.7. Policy concept ..... 17
        1.2.8. Listener/notifier replication ..... 17
        1.2.9. Virtualization and cloud management ..... 17
    1.3. Further documentation ..... 18
    1.4. Symbols and conventions used in this manual ..... 18
2. Installation ..... 21
    2.1. Introduction ..... 21
    2.2. Selecting the installation mode ..... 22
    2.3. Selecting the installation language ..... 23
    2.4. Selecting the location ..... 23
    2.5. Selecting the keyboard layout ..... 24
    2.6. Network configuration ..... 25
    2.7. Setting up the root password ..... 27
    2.8. Partitioning the hard drive ..... 27
    2.9. Domain settings ..... 29
        2.9.1. "Create a new UCS domain" mode ..... 30
        2.9.2. "Join an existing Active Directory domain" mode ..... 31
        2.9.3. "Join an existing UCS domain domain" mode ..... 32
        2.9.4. "Do not use any domain" mode ..... 33
    2.10. Selecting UCS software components ..... 33
    2.11. Confirming the settings ..... 34
    2.12. Troubleshooting for installation problems ..... 35
    2.13. Installation in text mode ..... 35
    2.14. Installation in the Amazon EC2 cloud ..... 36
    2.15. Installation in VMware ..... 36
    2.16. Installation as Docker image ..... 36
    2.17. Installation in Citrix XenServer ..... 36
3. Domain services / LDAP directory ..... 37
    3.1. Introduction ..... 38
    3.2. Joining domains ..... 38
        3.2.1. How UCS systems join domains ..... 38
            3.2.1.1. Subsequent domain joins with `univention-join` ..... 39
            3.2.1.2. Joining domains with Univention Management Console ..... 39
            3.2.1.3. Join scripts / Unjoin scripts ..... 39
        3.2.2. Windows domain joins ..... 40
            3.2.2.1. Windows 10 ..... 41
            3.2.2.2. Windows 8 ..... 41
            3.2.2.3. Windows 7 ..... 42
            3.2.2.4. Windows Server 2012 ..... 42
        3.2.3. Ubuntu domain joins ..... 42
        3.2.4. Mac OS X domain joins ..... 42
            3.2.4.1. Domain join using the system preferences GUI ..... 42
            3.2.4.2. Domain join on the command line ..... 43

3.3. UCS system roles ..... 43
3.3.1. Domain controller master ..... 43
3.3.2. Domain controller backup ..... 44
3.3.3. Domain controller slave ..... 44
3.3.4. Member server ..... 44
3.3.5. Base system ..... 44
3.3.6. Ubuntu ..... 44
3.3.7. Linux ..... 44
3.3.8. Univention Corporate Client ..... 44
3.3.9. Mac OS X ..... 44
3.3.10. Domain Trust Account ..... 45
3.3.11. IP managed client ..... 45
3.3.12. Windows Domaincontroller ..... 45
3.3.13. Windows Workstation/Server ..... 45
3.4. LDAP directory ..... 45
3.4.1. LDAP schemas ..... 45
3.4.1.1. LDAP schema extensions ..... 45
3.4.1.2. LDAP schema replication ..... 45
3.4.2. Audit-proof logging of LDAP changes ..... 46
3.4.3. Timeout for inactive LDAP connections ..... 47
3.4.4. LDAP command line tools ..... 47
3.4.5. Access control for the LDAP directory ..... 47
3.4.5.1. Delegation of the privilege to reset user passwords ..... 48
3.4.6. Name Service Switch / LDAP NSS module ..... 49
3.4.7. Syncrepl for synchronization with non-UCS OpenLDAP servers ..... 49
3.4.8. Configuration of the directory service when using Samba 4 ..... 49
3.4.9. Daily backup of LDAP data ..... 49
3.5. Listener/notifier domain replication ..... 49
3.5.1. Listener/notifier replication workflow ..... 49
3.5.2. Analysis of listener/notifier problems ..... 51
3.5.2.1. Log files/debug level of replication ..... 51
3.5.2.2. Identification of replication problems ..... 51
3.5.2.3. Reinitialization of listener modules ..... 51
3.6. SSL certificate management ..... 51
3.7. Kerberos ..... 53
3.8. Password hashes in the directory service ..... 53
3.9. SAML identity provider ..... 54
3.9.1. Login via *single sign-on* ..... 55
3.9.2. Adding a new external service provider ..... 55
3.9.3. Extended Configuration ..... 57
3.10. OpenID Connect Provider ..... 57
3.11. Converting a backup domain controller backup to the new master domain controller ..... 59
3.12. Fault-tolerant domain setup ..... 60
3.13. Protocol of activities in the domain ..... 61
4. UCS web interface ..... 63
4.1. Introduction ..... 64
4.1.1. Access ..... 65
4.1.2. Browser compatibility ..... 65
4.1.3. Feedback on UMC and UCS ..... 65
4.1.4. Collection of usage statistics ..... 66
4.2. Login ..... 66
4.3. UCS portal page ..... 67
4.3.1. Assign rights for portal settings ..... 68
4.4. Univention Management Console ..... 69

4.4.1. Introduction ..... 69
4.4.2. Activation of UCS license / license overview ..... 69
4.4.3. Operating instructions for modules to administrate LDAP directory data ..... 70
4.4.3.1. Searching for objects ..... 72
4.4.3.2. Creating objects ..... 73
4.4.3.3. Editing objects ..... 73
4.4.3.4. Deleting objects ..... 73
4.4.3.5. Moving objects ..... 73
4.4.4. Favorites ..... 73
4.4.5. Display of system notifications ..... 73
4.5. LDAP directory browser ..... 73
4.6. Policies ..... 75
4.6.1. Creating a policy ..... 75
4.6.2. Applying policies ..... 75
4.6.3. Editing a policy ..... 76
4.7. Expansion of UMC with extended attributes ..... 76
4.8. Structuring of the domain with user-defined LDAP structures ..... 80
4.9. Delegated administration in the UMC ..... 81
4.10. Command line interface of domain management (Univention Directory Manager) ..... 81
4.10.1. Parameters of the command line interface ..... 82
4.10.2. Example invocations of the command line interface ..... 84
4.10.2.1. Users ..... 84
4.10.2.2. Groups ..... 85
4.10.2.3. Container / Policies ..... 85
4.10.2.4. Computers ..... 86
4.10.2.5. Shares ..... 86
4.10.2.6. Printers ..... 86
4.10.2.7. DNS/DHCP ..... 87
4.10.2.8. Extended attributes ..... 87
4.11. HTTP API of domain management ..... 88
4.12. Evaluation of data from the LDAP directory with Univention Directory Reports ..... 88
4.12.1. Creating reports in Univention Management Console ..... 88
4.12.2. Creating reports on the command line ..... 89
4.12.3. Adjustment/expansion of Univention Directory Reports ..... 89
5. Software deployment ..... 91
5.1. Introduction ..... 91
5.2. Differentiation of update variants / UCS versions ..... 91
5.3. Univention App Center ..... 92
5.4. Updates of UCS systems ..... 96
5.4.1. Update strategy in environments with more than one UCS system ..... 96
5.4.2. Updating individual systems via Univention Management Console ..... 97
5.4.3. Updating individual systems via the command line ..... 98
5.4.4. Updating systems via a policy ..... 98
5.4.5. Postprocessing of release updates ..... 99
5.4.6. Troubleshooting in case of update problems ..... 99
5.5. Configuration of the repository server for updates and package installations ..... 99
5.5.1. Configuration via Univention Management Console ..... 99
5.5.2. Configuration via Univention Configuration Registry ..... 99
5.5.3. Policy-based configuration of the repository server ..... 100
5.5.4. Creating and updating a local repository ..... 100
5.6. Installation of further software ..... 100
5.6.1. Installation/uninstallation of UCS components in the Univention App Center ..... 101
5.6.2. Installation/removal of individual packages in Univention Management Console ..... 101
5.6.3. Installation/removal of individual packages in the command line ..... 102

5.6.4. Hook scripts for administrators ..... 103
5.6.5. Policy-based installation/uninstallation of individual packages via package lists ..... 103
5.7. Specification of an update point using the package maintenance policy ..... 104
5.8. Central monitoring of software installation statuses with the software monitor ..... 104
6. User management ..... 107
6.1. User management with Univention Management Console ..... 107
6.2. User activation for apps ..... 114
6.3. User password management ..... 115
6.4. Password settings for Windows clients when using Samba ..... 116
6.5. User self services ..... 117
6.5.1. Password change by user via Univention Management Console ..... 117
6.5.2. Password management via Self Service app ..... 117
6.5.3. Contact information ..... 118
6.5.4. Self registration ..... 120
6.5.4.1. Account creation ..... 121
6.5.4.2. Verification email ..... 122
6.5.4.3. Account verification ..... 123
6.5.5. Self *deregistration* ..... 124
6.6. Automatic lockout of users after failed login attempts ..... 125
6.6.1. Samba Active Directory Service ..... 125
6.6.2. PAM-Stack ..... 126
6.6.3. OpenLDAP ..... 126
6.7. User templates ..... 127
6.8. Overlay module for recording an account's last successful LDAP bind ..... 128
7. Group management ..... 131
7.1. Managing groups in Univention Management Console ..... 131
7.2. Nested groups ..... 134
7.3. Local group cache ..... 134
7.4. Synchronization of Active Directory groups when using Samba 4 ..... 135
7.5. Overlay module for displaying the group information on user objects ..... 135
8. Computer management ..... 137
8.1. Management of computer accounts in Univention Management Console ..... 138
8.1.1. Integration of Ubuntu clients ..... 142
8.2. Configuration of hardware and drivers ..... 142
8.2.1. Available kernel variants ..... 142
8.2.2. Hardware drivers / kernel modules ..... 143
8.2.3. GRUB boot manager ..... 143
8.2.4. Network configuration ..... 145
8.2.4.1. Network interfaces ..... 145
8.2.4.2. Configuring proxy access ..... 149
8.2.5. Configuration of the monitor settings ..... 149
8.2.6. Mounting NFS shares ..... 150
8.2.7. Collection of list of supported hardware ..... 150
8.3. Administration of local system configuration with Univention Configuration Registry ..... 151
8.3.1. Introduction ..... 151
8.3.2. Using the Univention Management Console web interface ..... 152
8.3.3. Using the command line front end ..... 152
8.3.3.1. Querying a UCR variable ..... 152
8.3.4. Policy-based configuration of UCR variables ..... 154
8.3.5. Modifying UCR templates ..... 155
8.3.5.1. Referencing of UCR variables in templates ..... 155
8.3.5.2. Integration of inline Python code in templates ..... 155
8.4. Basic system services ..... 156
8.4.1. Administrative access with the root account ..... 156

8.4.2. Configuration of language and keyboard settings ..... 156
8.4.3. Starting/stopping system services / configuration of automatic startup ..... 157
8.4.4. Authentication / PAM ..... 158
8.4.4.1. Limiting authentication to selected users ..... 158
8.4.5. Configuration of the LDAP server in use ..... 159
8.4.6. Configuration of the print server in use ..... 159
8.4.7. Logging/retrieval of system messages and system status ..... 159
8.4.7.1. Log files ..... 159
8.4.7.2. Logging the system status ..... 159
8.4.7.3. Querying system statistics in Univention Management Console ..... 160
8.4.7.4. Process overview in Univention Management Console ..... 160
8.4.7.5. System error diagnosis in Univention Management Console ..... 161
8.4.8. Executing recurring actions with Cron ..... 161
8.4.8.1. Hourly/daily/weekly/monthly execution of scripts ..... 161
8.4.8.2. Defining local cron jobs in `/etc/cron.d/` ..... 161
8.4.8.3. Defining cron jobs in Univention Configuration Registry ..... 162
8.4.9. Name service cache daemon ..... 162
8.4.10. RDP login to systems using XRDP ..... 163
8.4.10.1. Installation ..... 163
8.4.10.2. Configuration ..... 163
8.4.10.3. Client software ..... 164
8.4.10.4. Known issue: Wrong keyboard layout ..... 164
8.4.10.5. Alternatives ..... 164
8.4.11. SSH login to systems ..... 164
8.4.12. Configuring the time zone / time synchronization ..... 165
9. Services for Windows ..... 167
9.1. Introduction ..... 167
9.2. Operation of a Samba domain based on Active Directory ..... 168
9.2.1. Installation ..... 168
9.2.2. Services of a Samba domain ..... 168
9.2.2.1. Authentication services ..... 168
9.2.2.2. File services ..... 169
9.2.2.3. Print services ..... 169
9.2.2.4. Univention S4 connector ..... 169
9.2.2.5. Replication of directory data ..... 170
9.2.2.6. Synchronization of the SYSVOL share ..... 170
9.2.3. Configuration and management of Windows desktops ..... 170
9.2.3.1. Group policies ..... 170
9.2.3.2. Logon scripts / NETLOGON share ..... 176
9.2.3.3. Configuration of the file server for the home directory ..... 176
9.2.3.4. Roaming profiles ..... 176
9.3. Active Directory Connection ..... 177
9.3.1. Introduction ..... 177
9.3.2. UCS as a member of an Active Directory domain ..... 177
9.3.3. Setup of the UCS AD connector ..... 179
9.3.3.1. Basic configuration of the UCS AD Connector ..... 180
9.3.3.2. Importing the SSL certificate of the Active Directory ..... 182
9.3.3.3. Starting/Stopping the Active Directory Connection ..... 184
9.3.3.4. Functional test of basic settings ..... 184
9.3.3.5. Changing the AD access password ..... 184
9.3.4. Additional tools / Debugging connector problems ..... 185
9.3.4.1. `univention-adsearch` ..... 185
9.3.4.2. `univention-connector-list-rejected` ..... 185
9.3.4.3. Logfiles ..... 185

9.3.5. Details on preconfigured synchronization ..... 185
9.3.5.1. Containers and organizational units ..... 185
9.3.5.2. Groups ..... 186
9.3.5.3. Users ..... 187
9.4. Migrating an Active Directory domain to UCS using Univention AD Takeover ..... 187
9.4.1. Introduction ..... 187
9.4.2. Preparation ..... 188
9.4.3. Domain migration ..... 188
9.4.4. Final steps of the takeover ..... 191
9.4.5. Tests ..... 191
9.5. Trust relationships ..... 191
10. Identity Management connection to cloud services ..... 195
10.1. Introduction ..... 195
10.2. Microsoft 365 Connector ..... 195
10.2.1. Setup ..... 195
10.2.2. Configuration ..... 196
10.2.2.1. Users ..... 196
10.2.2.2. Teams ..... 197
10.2.3. Synchronization of Users in multiple Azure Active Directories ..... 197
10.2.4. Troubleshooting/Debugging ..... 198
10.3. Google Apps for Work Connector ..... 198
10.3.1. Setup ..... 198
10.3.2. Configuration ..... 199
10.3.3. Troubleshooting/Debugging ..... 200
11. IP and network management ..... 201
11.1. Network objects ..... 202
11.2. Administration of DNS data with BIND ..... 203
11.2.1. Configuration of the BIND name server ..... 204
11.2.1.1. Configuration of BIND debug output ..... 204
11.2.1.2. Configuration of the data backend ..... 204
11.2.1.3. Configuration of zone transfers ..... 205
11.2.2. Administration of DNS data in Univention Management Console ..... 205
11.2.2.1. Forward lookup zone ..... 205
11.2.2.2. CNAME record (Alias records) ..... 207
11.2.2.3. A/AAAA records (host records) ..... 207
11.2.2.4. Service records ..... 208
11.2.2.5. Reverse lookup zone ..... 209
11.2.2.6. Pointer record ..... 209
11.3. IP assignment via DHCP ..... 210
11.3.1. Introduction ..... 210
11.3.2. Composition of the DHCP configuration via DHCP LDAP objects ..... 211
11.3.2.1. Administration of DHCP services ..... 211
11.3.2.2. Administration of DHCP server entries ..... 211
11.3.2.3. Administration of DHCP subnets ..... 212
11.3.2.4. Administration of DHCP pools ..... 212
11.3.2.5. Registration of computers with DHCP computer objects ..... 213
11.3.2.6. Management of DHCP shared networks / DHCP shared subnets ..... 214
11.3.3. Configuration of clients via DHCP policies ..... 214
11.3.3.1. Setting the gateway ..... 215
11.3.3.2. Setting the DNS servers ..... 215
11.3.3.3. Setting the WINS server ..... 215
11.3.3.4. Configuration of the DHCP lease ..... 216
11.3.3.5. Configuration of boot server/PXE settings ..... 216
11.3.3.6. Further DHCP policies ..... 216

11.4. Packet filter with Univention Firewall ..... 217
11.5. Web proxy for caching and policy management / virus scan ..... 217
11.5.1. Installation ..... 217
11.5.2. Caching of web content ..... 218
11.5.3. Logging proxy accesses ..... 218
11.5.4. Restriction of access to permitted networks ..... 218
11.5.5. Configuration of the ports used ..... 218
11.5.5.1. Access port ..... 218
11.5.5.2. Permitted ports ..... 219
11.5.6. User authentication on the proxy ..... 219
11.5.7. Filtering/policy enforcement of web content with DansGuardian ..... 219
11.5.8. Definition of content filters for DansGuardian ..... 220
11.6. RADIUS ..... 222
11.6.1. Installation ..... 222
11.6.2. Configuration ..... 222
11.6.2.1. Allowed users ..... 222
11.6.2.2. MAC filtering ..... 222
11.6.2.3. Access point administration ..... 223
11.6.2.4. Access point and client configuration ..... 224
11.6.3. Debugging ..... 224
12. File share management ..... 227
12.1. Access rights to data in shares ..... 227
12.2. Management of shares in UMC ..... 228
12.3. Support for MSDFS ..... 235
12.4. Configuration of file system quota ..... 235
12.4.1. Activating filesystem quota ..... 236
12.4.2. Configuring filesystem quota ..... 236
12.4.3. Evaluation of quota during login ..... 237
12.4.4. Querying the quota status by administrators or users ..... 237
13. Print services ..... 239
13.1. Introduction ..... 239
13.2. Installing a print server ..... 239
13.3. Setting the local configuration properties of a print server ..... 240
13.4. Creating a printer share ..... 240
13.5. Creating a printer group ..... 243
13.6. Administration of print jobs and print queues ..... 245
13.7. Generating PDF documents from print jobs ..... 246
13.8. Mounting of print shares in Windows clients ..... 247
13.9. Integrating additional PPD files ..... 251
14. Mail services ..... 253
14.1. Introduction ..... 253
14.2. Installation ..... 254
14.3. Management of the mail server data ..... 254
14.3.1. Management of mail domains ..... 254
14.3.2. Assignment of e-mail addresses to users ..... 255
14.3.3. Management of mailing lists ..... 255
14.3.4. Management of mail groups ..... 256
14.3.5. Management of shared IMAP folders ..... 257
14.3.6. Mail quota ..... 258
14.4. Spam detection and filtering ..... 259
14.5. Identification of viruses and malware ..... 260
14.6. Identification of Spam sources with *DNS-based Blackhole Lists* (DNSBL) ..... 260
14.7. Integration of Fetchmail for retrieving mail from external mailboxes ..... 261
14.8. Configuration of the mail server ..... 261

14.8.1. Configuration of a relay host for sending the e-mails ..... 261
14.8.2. Configuration of the maximum mail size ..... 262
14.8.3. Configuration of a blind carbon copy for mail archiving solutions ..... 262
14.8.4. Configuration of soft bounces ..... 262
14.8.5. Configuration of SMTP ports ..... 262
14.8.6. Configuration of additional checks by postscreen ..... 263
14.8.7. Custom Postfix configuration ..... 263
14.8.8. Configuring the Alias-expansion-limit ..... 264
14.8.9. Handling of mailboxes during e-mail changes and the deletion of user accounts ..... 264
14.8.10. Distribution of an installation on several mail servers ..... 265
14.8.11. Mail storage on NFS ..... 265
14.8.12. Connection limits ..... 265
14.9. Configuration of mail clients for the mail server ..... 267
14.10. Webmail and administration of e-mail filters with Horde ..... 267
14.10.1. Login and overview ..... 267
14.10.2. Web-based mail access ..... 268
14.10.3. Address book ..... 269
14.10.4. E-mail filters ..... 269
15. Infrastructure monitoring ..... 271
15.1. Introduction ..... 271
15.1.1. UCS Dashboard ..... 271
15.1.1.1. Introduction and structure ..... 271
15.1.1.2. Installation ..... 271
15.1.1.3. Usage ..... 272
15.2. Nagios ..... 273
15.2.1. Introduction and structure ..... 273
15.2.2. Installation ..... 275
15.2.2.1. Preconfigured Nagios checks ..... 275
15.2.3. Configuration of the Nagios monitoring ..... 277
15.2.3.1. Configuration of a Nagios service ..... 277
15.2.3.2. Configuration of a monitoring time period ..... 280
15.2.3.3. Assignment of Nagios checks to computers ..... 280
15.2.3.4. Integration of additional Nagios plugin configurations ..... 282
15.2.4. Querying the system status via the Nagios web interface ..... 282
15.2.5. Integration of additional plugins ..... 282
16. Virtualization ..... 285
16.1. Introduction ..... 285
16.2. Installation ..... 285
16.3. Creating connections to cloud computing instances ..... 286
16.3.1. Creating an OpenStack connection ..... 287
16.3.2. Creating an EC2 connection ..... 289
16.4. Managing virtual machines with Univention Management Console ..... 289
16.4.1. Operations (Starting/stopping/suspending/deleting/migrating/cloning virtual machines) ..... 290
16.4.2. Creating a virtual machine via a cloud connection ..... 292
16.4.3. Editing a virtual machine via a cloud connection ..... 292
16.4.4. Creating a virtual instance ..... 292
16.4.5. Modifying virtual machines ..... 293
16.5. KVM related UVMM features ..... 295
16.5.1. Image files of virtual machines ..... 295
16.5.2. Storage pools ..... 296
16.5.2.1. Accessing the default storage pool through a file share ..... 296
16.5.2.2. Adding a storage pool ..... 297
16.5.2.3. Moving the default storage pool ..... 297

16.5.3. CD/DVD/floppy drives in virtual machines ..... 298
16.5.4. Network interfaces in virtual instances ..... 298
16.5.5. Paravirtualization (virtIO) drivers for Microsoft Windows systems ..... 299
16.5.5.1. Installation of the virtIO drivers for KVM instances ..... 299
16.5.6. Snapshots ..... 299
16.5.7. Migration of virtual instances ..... 299
16.5.7.1. Migration of virtual machines from failed virtualization servers ..... 300
16.5.7.2. Migration of virtual machines between hosts with different CPUs ..... 300
16.6. Profiles ..... 301
16.6.1. Changing default network ..... 301
17. Data backup with Bacula ..... 303
17.1. Introduction ..... 303
17.2. Scope of backup on a UCS system ..... 304
17.3. Installation ..... 304
17.4. Configuration of the backup components ..... 305
17.4.1. Directory Daemon ..... 305
17.4.2. Storage ..... 305
17.4.3. File Daemon ..... 305
17.4.4. Bacula Console ..... 306
17.4.5. Firewall adjustments ..... 306
17.5. Configuration of the backup (interval, data, etc.) ..... 306
17.6. Administration via the Bacula console ..... 307
17.7. Backup of the catalog database ..... 308
17.8. Further information ..... 309
Bibliography ..... 311

# Chapter 1. Introduction

1.1. What is Univention Corporate Server? ..... 13
1.2. Overview of UCS ..... 14
1.2.1. Commissioning ..... 14
1.2.2. Domain concept ..... 14
1.2.3. Expandability with the Univention App Center ..... 15
1.2.4. LDAP directory service ..... 15
1.2.5. Domain administration ..... 16
1.2.6. Computer administration ..... 17
1.2.7. Policy concept ..... 17
1.2.8. Listener/notifier replication ..... 17
1.2.9. Virtualization and cloud management ..... 17
1.3. Further documentation ..... 18
1.4. Symbols and conventions used in this manual ..... 18

## 1.1. What is Univention Corporate Server?

Feedback

Univention Corporate Server (UCS) is a Linux-based server operating system for the operation and administration of IT infrastructures for companies and authorities. UCS implements an integrated, holistic concept with consistent, central administration and can ensure the operation of all the components in an interrelated security and trust context, the so-called UCS domain. At the same time, UCS supports a wide range of open standards and includes extensive interfaces to infrastructure components and management tools from other manufacturers, meaning it can be easily integrated in existing environments.

UCS consists of reliable Open Source software tried and tested in organizations of different sizes. These software components are integrated together via the UCS management system. This allows the easy integration and administration of the system in both simple and complex distributed or virtualized environments.

The central functions of UCS are:

- Flexible and extensive identity/infrastructure management for the central administration of servers, workstations, users and their permissions, server applications and web services
- Services for integrating the management of existing Microsoft Active Directory domains or even the provision of such services as an alternative for Microsoft-based server systems
- App Center for simple installation and management of extensions and applications
- Comprehensive features for the operation of virtualized systems (e.g. running a Windows or Linux operating systems) in either the cloud of on locally running UCS systems
- Network and intranet services for administration of DHCP and DNS
- File and print services
- Computer administration and monitoring
- Mail services

These functions are provided by different software packages in Univention Corporate Server and are handled in detail in the course of this handbook. Basically, the software packages contained in UCS can be assigned to the following three main categories:

1. Base system

2. UCS management system with Univention Management Console

3. Univention App Center, allowing the installation of further components and applications of other software vendors

The *base system* encompasses the operating system of the UCS Linux distribution maintained by Univention and based on Debian GNU/Linux. It largely includes the same software selection as Debian GNU/Linux as well as additional tools for the installation, updating and configuration of clients and servers.

The *UCS management system* realizes a single point of administration where the accounts of all domain members (users, groups, and hosts) and services such as DNS and DHCP are managed in a single directory service. Core components of the management system are the services OpenLDAP (directory service), Samba (provision of domain, file and print services for Windows), Kerberos (authentication and single sign on), DNS (network name resolution) and SSL/TLS (secure transmission of data between systems). It can be used either via a web interface (Univention Management Console) or in the command line and in individual scripts. The UCS management system can be extended with APIs (application programming interfaces) and provides a flexible client-server architecture which allows changes to be transferred to the involved systems and be activated there.

Additional components from Univention and other manufacturers can easily be installed using the App Center. They expand the system with numerous functions such as groupware, document management and services for Windows, meaning that they can also be run from a UCS system and administrated via the UCS management system.

## 1.2. Overview of UCS

Feedback

Linux is an operating system which always had a focus on stability, security and compatibility with other operating systems. Therefore Linux is predestined for being used in server operating systems that are stable, secure and highly available.

Built on that base, UCS is a server operating system which is optimized for the simple and secure operation and management of applications and infrastructure services in enterprises and public authorities. For efficient and secure management such applications rely on the tight integration in the user and permission management of the UCS management system.

UCS can be employed as the basis for the IT infrastructure in companies and authorities and provide the central control for it. This makes a considerable contribution to secure, efficient and cost-effective IT operation. The business-critical applications are integrated in a uniform concept, adapted to each other and pre-configured for professional utilization. Alternatively it can be operated as part of an existing Microsoft Active Directory domain.

### 1.2.1. Commissioning

Feedback

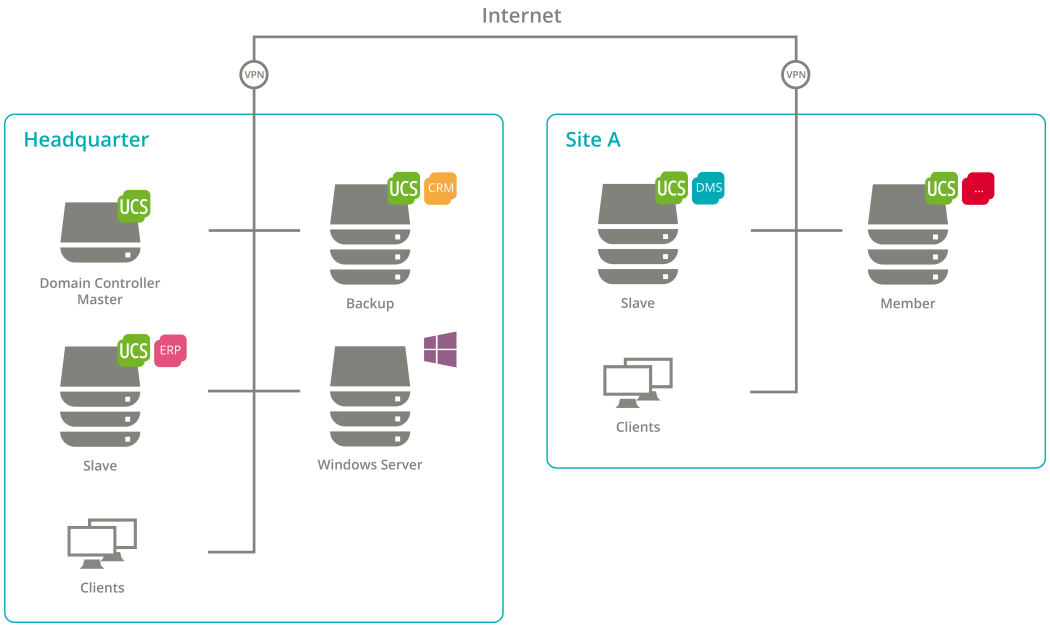
The use of UCS begins either with a classic operating system installation on a physical server or as a virtual machine. Further information can be found in Chapter 2.

### 1.2.2. Domain concept

Feedback

In an IT infrastructure managed with UCS, all servers, clients and users are contained in a common security and trust context, referred to as the UCS domain. Every UCS system is assigned a so-called server role during the installation. Possible system roles are domain controller, member server and client.

**Figure 1.1. UCS domain concept**




Depending on the system role within the domain, such services as Kerberos, OpenLDAP, Samba, modules for domain replication or a Root CA (certification authority) are installed on the computer. These are automatically configured for the selected system role. The manual implementation and configuration of every single service and application is therefore not required. Due to the modular design and extensive configuration interfaces, tailor-made solutions to individual requirements can nevertheless be realized.

The integration of Samba, which provides the domain service for clients and servers operated with Microsoft Windows, makes Univention Corporate Server compatible with Microsoft Active Directory (AD), whereby the system acts as an Active Directory server for Windows-based systems. Consequently, for example, group policies for Microsoft Windows systems can be administrated in the usual way.

UCS can also be operated as part of an existing Microsoft Active Directory domain. This way, users and groups of the Active Directory domain can access applications from the Univention App Center.

Ubuntu or Mac OS X clients can be integrated in a UCS environment, as well (see Section 8.1.1).

## 1.2.3. Expandability with the Univention App Center

Feedback

The Univention App Center offers additional UCS components and extensions and a broad selection of business IT software, e.g., groupware and collaboration, file exchange, CRM or backup. These applications can be installed in existing environments with a few clicks and are usually ready to use. In most cases they are directly integrated into the UCS management system such that they are available in Univention Management Console. This provides a central management of data on the domain level and obsoletes the separate management of, e.g., user data in multiple places.

## 1.2.4. LDAP directory service

Feedback

With the UCS management system, all the components of the UCS domain can be centrally administrated across computer, operating system and site boundaries. It thus provides a single point of administration for the domain. One primary element of the UCS management system is an LDAP directory in which the data required across the domain for the administration are stored. In addition to the user accounts and similar elements, the data basis of services such as DHCP is also saved there. The central data management in the

LDAP directory avoids not only the repeated entry of the same data, but also reduces the probability of errors and inconsistencies.

An LDAP directory has a tree-like structure, the root of which forms the so-called basis of the UCS domain. The UCS domain forms the common security and trust context for its members. An account in the LDAP directory establishes the membership in the UCS domain for users. Computers receive a computer account when they join the domain. Microsoft Windows systems can also join the domain such that users can log in there with their domain passport.

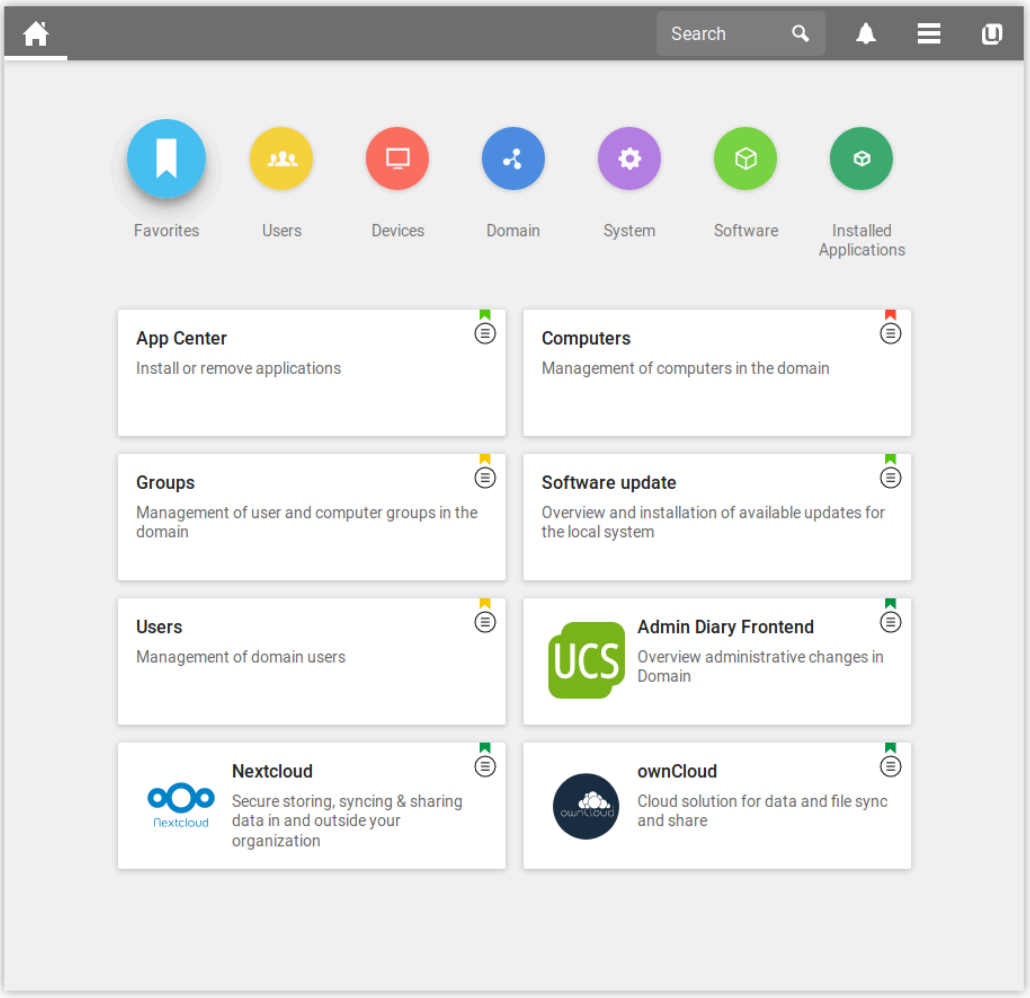
UCS utilizes OpenLDAP as a directory service server. The directory is provided by the master domain controller and replicated on all domain controllers (DCs) in the domain. The complete LDAP directory is also replicated on a DC backup as this can replace the DC master in an emergency. In contrast, the replication on DC slaves can be restricted to certain areas of the LDAP directory using ACLs (access control lists) in order to realize a selective replication. For example, this may be desirable if data should only be stored on as few servers as possible for security reasons. For secure communication of all systems within the domain, UCS integrates a root CA (certification authority).

Further information can be found in Section 3.4.

## 1.2.5. Domain administration

Feedback

**Figure 1.2. Univention Management Console**



Access to the LDAP directory is performed via the web-based user interface Univention Management Console (UMC). In addition to this, Univention Directory Manager allows the realization of all domain-wide administrative tasks via a command line interface. This is particularly suitable for the integration in scripts or automated administrative steps.

Univention Management Console allows to display, edit, delete, and search the data in the LDAP directory via various filter criteria. The web interface offers a range of wizards for the administration of user, groups, networks, computers, directory shares and printers. The administration of computers also comprises comprehensive functions for distributing and updating software. The integrated LDAP directory browser can be used to make further settings and add customer-specific object classes or attributes.

Further information can be found in Chapter 4.

## 1.2.6. Computer administration

Feedback

Univention Management Console allows not only the access to the LDAP directory, but also the web-based configuration and administration of individual computers. These include the adaptation of configuration data, the installation of software as well as the monitoring and control of services and the operating system itself. With the UCS management system, domain administration as well as computer and server configuration is possible from any place via a comfortable graphic web interface.

## 1.2.7. Policy concept

Feedback

The tree-like structure of LDAP directories is similar to that of a file system It ensures that objects (such as users, computers, etc.) are in one container which itself can be adopted by other containers. The root container is also called the LDAP base object.

Policies describe certain administrative settings which are applied to more than one object. Linked to containers, they facilitate the administration as they are effective for all objects in the container in question as well as the objects in subfolders.

For example, users can be organized in different containers or organizational units (which are a form of containers) depending on which department they belong to. Settings such as the desktop background or accessible programs can then be connected to these organizational units using policies. Subsequently, they apply for all users within the organizational unit in question.

Further information can be found in Section 4.6.

## 1.2.8. Listener/notifier replication

Feedback

The listener/notifier mechanism is an important technical component of the UCS management system. With this, the creation, editing or deleting of entries in the LDAP directory triggers defined actions on the computers in question. For example, the creation of a directory share with Univention Management Console leads to the share firstly being entered in the LDAP directory. The listener/notifier mechanism then ensures that the NFS and Samba configuration files are also expanded accordingly on the selected server and that the directory is created in the file system of the selected server if it does not already exist.

The listener/notifier mechanism can be easily expanded with modules for further - also customer-specific - procedures. Consequently, it is used by numerous technology partners for the integration of their products in the LDAP directory service and the UCS management system for example.

Further information can be found in Section 3.5.

## 1.2.9. Virtualization and cloud management

Feedback

With the UMC module UCS Virtual Machine Manager (UVMM), UCS offers an extensive, powerful tool for the administration of hybrid cloud environments virtualization servers registered in the UCS domain and

virtual machines operated on it can be centrally monitored and administrated. In addition UVMM offers the possibility to manage virtual machines in OpenStack or EC2 environments.

Further information can be found in Chapter 16.

# 1.3. Further documentation

Feedback

This manual addresses just a small selection of the possibilities in UCS. Among other things, UCS and solutions based on UCS provide:

- Comprehensive support for complex server environments and replication scenarios
- Advanced capabilities for Windows environments
- Central network management with DNS and DHCP
- Monitoring systems and networks with Nagios
- Print server functionalities
- Thin Client support
- Fax service
- Proxy server
- Virtualization
- Integrated backup functions
- Linux desktop for business operations

Further documentation related to UCS and further issues is published under [ucs-dokumentationen] and in the Univention Wiki (https://wiki.univention.de/).

# 1.4. Symbols and conventions used in this manual

Feedback

The manual uses the following symbols:

### Caution

Warnings are highlighted.

### Note

Notes are also highlighted.

This tables describes the functionality of a UMC module:

**Table 1.1. Tab Nagios service**

| Attribute | Description |
|---|---|
| Name | The unique name of a Nagios service. |
| Description | An arbitrary description of the Nagios service. |

Menu entries, button labels, and similar details are printed in **bold** lettering. In addition, **[button labels]** are represented in square parentheses.

Names are in *bold*.

*Computer names, LDAP DNs*, `program names`, `file names, file paths`, `internet addresses` and *options* are also optically accented.

`Commands and other keyboard input` is printed in the Courier font.

```
In addition, excerpts from configuration files, screen output, etc are
printed on a grey background.
```

A backslash (\) at the end of a line signifies that the subsequent line feed is not to be understood as an *end of line*. This circumstance may occur, for example, where commands cannot be represented in one line in the manual, yet have to be entered in the command line in one piece without the backslash or with the backslash and a subsequent Enter.

The path to a function is represented in a similar way to a file path. **Users -> Add** means for example, you have to click **Users** in the main menu and **Add** in the submenu.

# Chapter 2. Installation

2.1. Introduction ..... 21
2.2. Selecting the installation mode ..... 22
2.3. Selecting the installation language ..... 23
2.4. Selecting the location ..... 23
2.5. Selecting the keyboard layout ..... 24
2.6. Network configuration ..... 25
2.7. Setting up the root password ..... 27
2.8. Partitioning the hard drive ..... 27
2.9. Domain settings ..... 29
2.9.1. "Create a new UCS domain" mode ..... 30
2.9.2. "Join an existing Active Directory domain" mode ..... 31
2.9.3. "Join an existing UCS domain domain" mode ..... 32
2.9.4. "Do not use any domain" mode ..... 33
2.10. Selecting UCS software components ..... 33
2.11. Confirming the settings ..... 34
2.12. Troubleshooting for installation problems ..... 35
2.13. Installation in text mode ..... 35
2.14. Installation in the Amazon EC2 cloud ..... 36
2.15. Installation in VMware ..... 36
2.16. Installation as Docker image ..... 36
2.17. Installation in Citrix XenServer ..... 36

## 2.1. Introduction

Feedback

The following documentation describes how to install Univention Corporate Server (UCS). The UCS system is installed from the DVD. The installation is interactive and prompts all the necessary system settings in a graphic interface.

The installation DVD is available for the computer architecture *amd64* (64-bit). In addition to support for the widely distributed BIOS systems, the DVD also includes support for the Unified Extensible Firmware Interface (UEFI) standard. The UEFI support on the DVD is also capable of starting systems with activated SecureBoot and installing UCS there.

Following installation on hardware or in a virtualization solution, UCS can also be installed on the Amazon EC2 cloud using an AMI image. Further information can be found in Section 2.14.

The installer's input masks can be operated with the mouse or via the keyboard.

- The **Tab** key can be used to proceed to the next field.
- The key combination of **Shift**+**Tab** can be used to return to the previous field.
- The **Enter** key is used to assign values to the input field and confirm buttons.
- Within a list or table, the *arrow keys* can be used for navigating between entries.

### Note

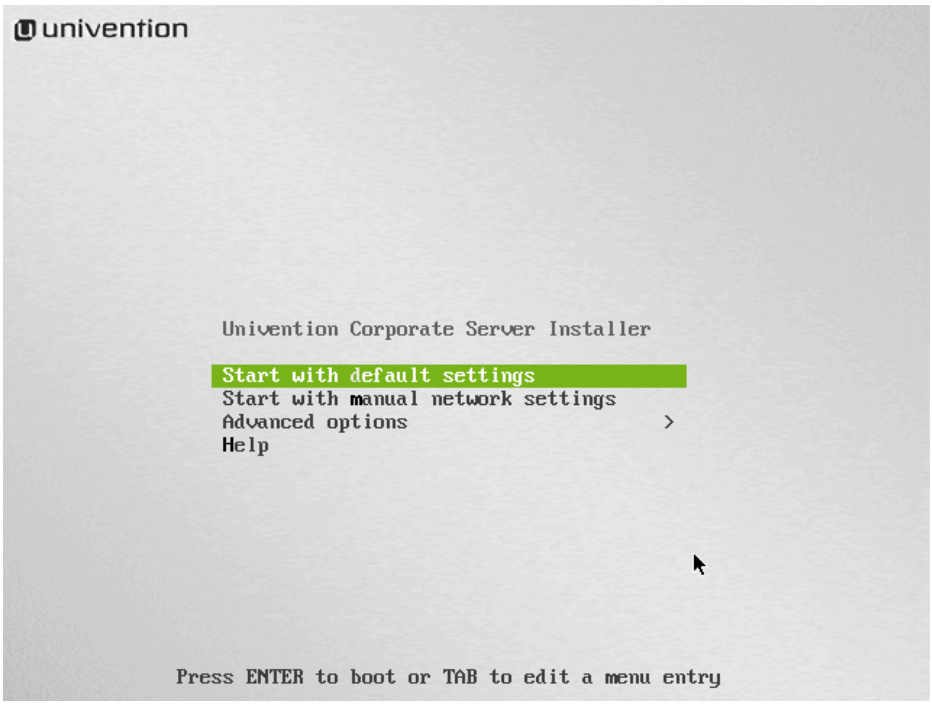
The **Cancel** button can be used to cancel the current configuration step. An earlier configuration step can then be selected again in the menu that is subsequently shown. Under certain circumstances, subsequent configuration steps cannot be directly selected if the earlier steps have not been completed.

# 2.2. Selecting the installation mode

Feedback

After booting the system from the installation medium, the following boot prompt is displayed:

**Figure 2.1. Installation boot prompt**




Now you can choose between several installation procedures.

- **Start with default settings** starts the interactive, graphic installation. During the installation, the system requests a number of parameters such as the network settings, hard drive partitions, domain settings and selection of software components for the UCS system to be installed and then performs the installation and the configuration.

- **Start with manual network settings** performs a standard installation, where the network is not configured automatically through DHCP. This is practical on systems, where the network must be setup manually.

- The **Advanced options** submenu offers advanced options for the installation process for selection:

  - **Install in text mode** performs an interactive standard installation in text mode. This is practical on systems which display problems with the graphic version of the installer.

  - **Boot from first hard drive** boots the operating system installed on the first hard drive instead of the UCS installation.

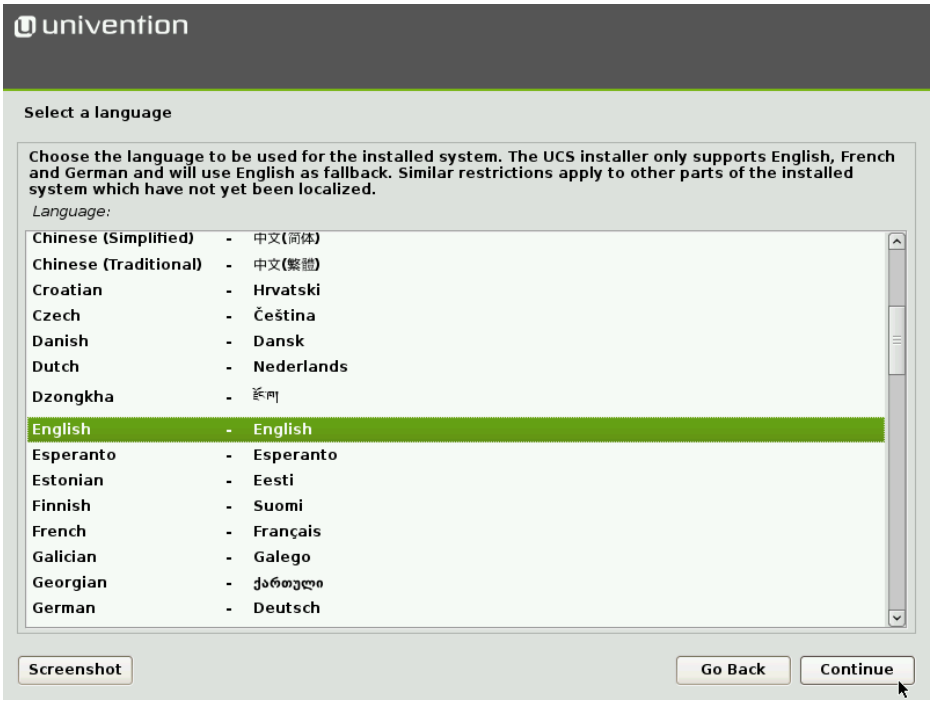
Once one of the installation option is selected, the kernel is loaded from the installation medium. The actual installation is divided into separate modules, which can be loaded from the installation medium subsequently if necessary. There are modules for network configuration or for selecting the software to be installed, among others.

## 2.3. Selecting the installation language

Feedback

In the first step, you can select the system language you wish to use. The selection has an influence on the use of language-specific characters and permits the representation of program output in the selected languages in the installed UCS system.

**Figure 2.2. Selecting the installation language**



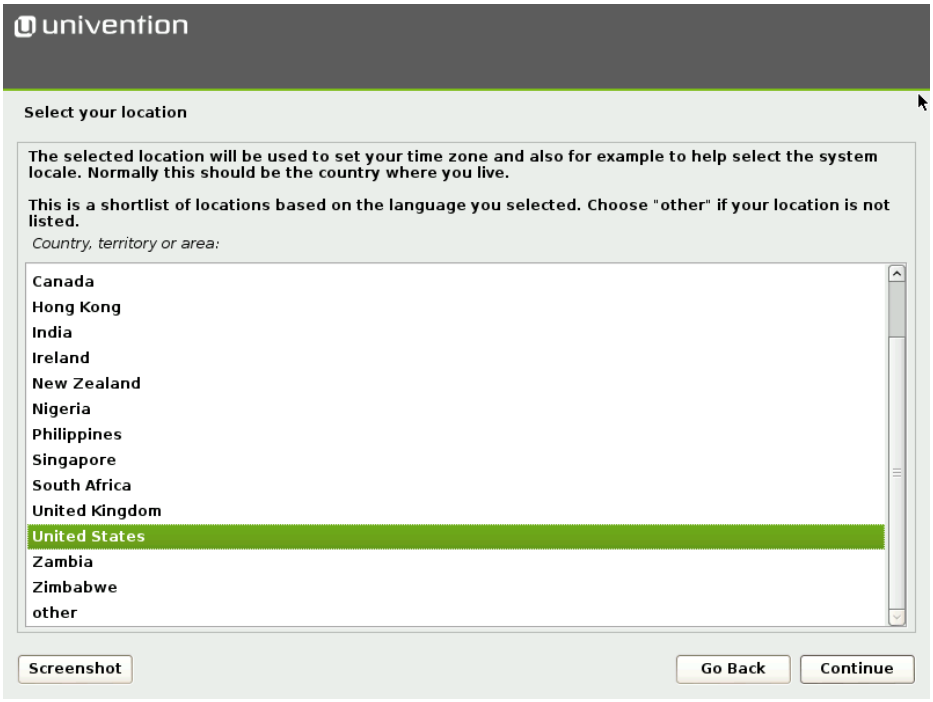

If Univention Installer has been translated into the selected language (currently German and English), the selected language is also used during the installation, otherwise the installation is performed in English.

## 2.4. Selecting the location

Feedback

Once the system language has been selected, a small list of locations is displayed based on the selected language. Select a suitable location from the list. The selected location is used to set the time zone or the correct language variant, for example. Should none of the displayed locations be appropriate, a more extensive list can be displayed using the menu entry **other**.

**Figure 2.3. Selecting the location**

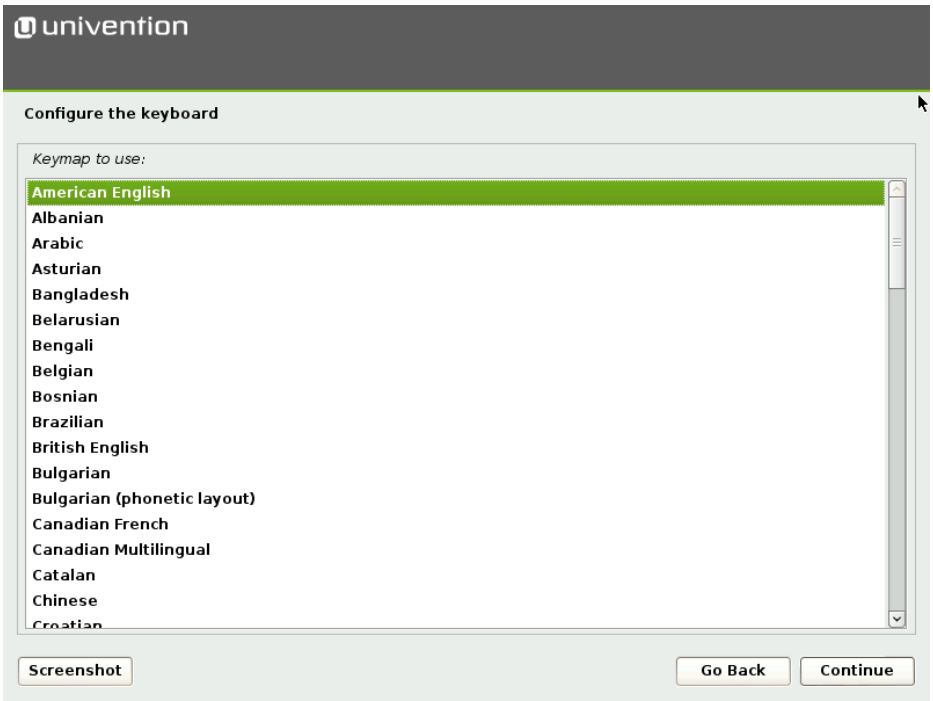



## 2.5. Selecting the keyboard layout

Feedback

The keyboard layout can be selected independently of the system language. The language selected here should be compatible with the keyboard used as it may otherwise cause operating problems.

**Figure 2.4. Selecting the keyboard layout**

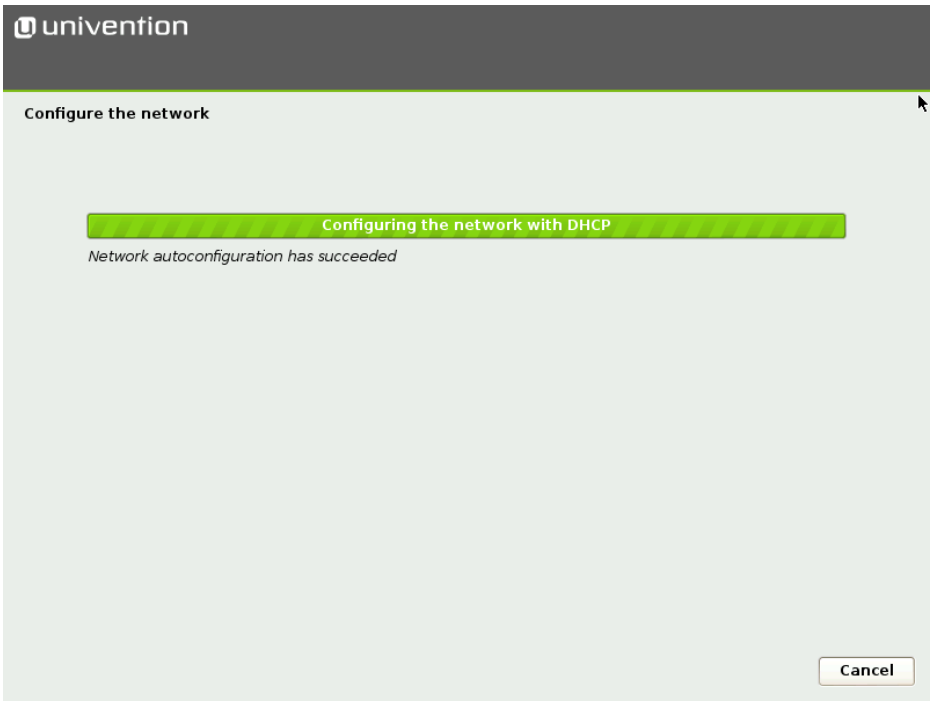


# 2.6. Network configuration

Feedback

Initially, the Univention Installer attempts to configure the network interfaces automatically. This can be disabled by selecting the menu item **Start with manual network settings** from the menu of the bootloader. Firstly, an attempt is made to determine an IPv6 address via the stateless address autoconfiguration (SLAAC). If this is not successful, the Univention Installer attempts to request an IPv4 address via the Dynamic Host Configuration Protocol (DHCP). If this is successful, the manual network configuration of Univention Installer is skipped.

**Figure 2.5. Automatic network configuration**

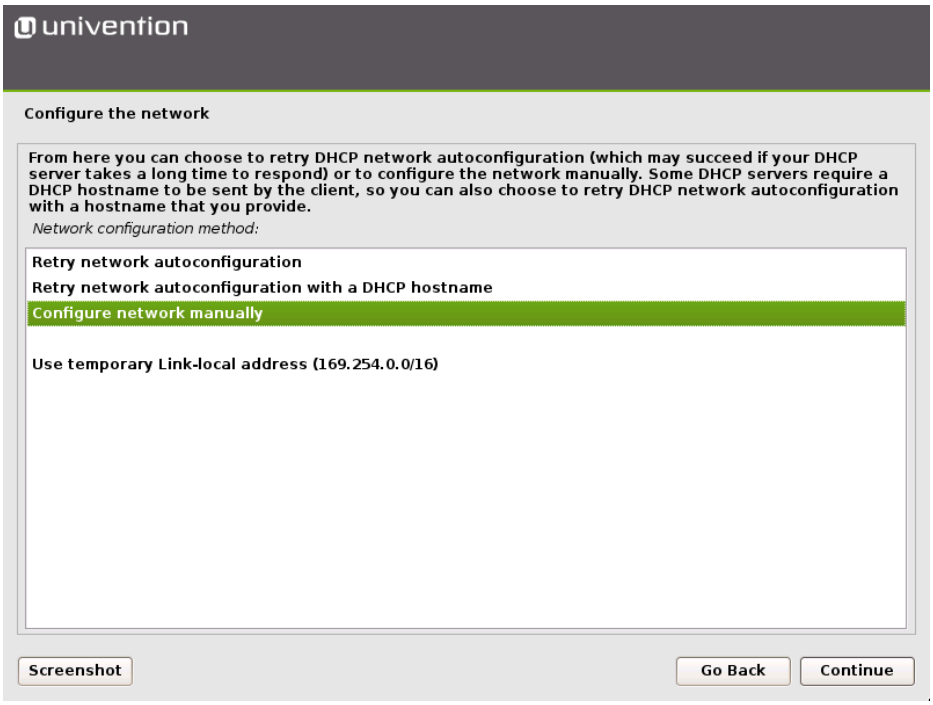



If there is no DHCP server present in the local network or static configuration of the network interface is required, the **Cancel** button can be selected. The Univention Installer then offers to repeat the automatic configuration or to configure the interface manually.

## Note

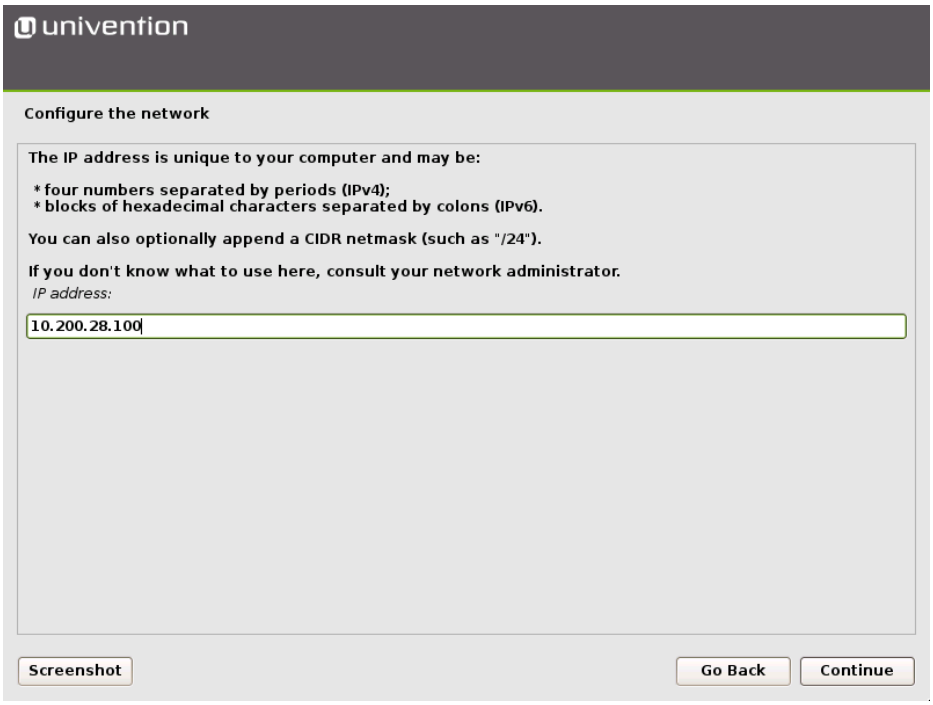
At least one network interface is required for the installation of Univention Corporate Server. If no supported network card is detected, Univention Installer opens a list of supported drivers for selection.

**Figure 2.6. Selecting the manual network configuration**




In manual configuration it is possible to specify either a static IPv4 or an IPv6 address for the system. IPv4 addresses have a 32-bit length and are generally written in four blocks in decimal form (e.g., `192.0.2.10`), whereas IPv6 addresses are four times as long and typically written in hexadecimal form (e.g., `2001:0DB8:FE29:DE27:0000:0000:0000:0000`). In addition to entering a static IP address, values for network masks, gateways and DNS servers are also requested.

**Figure 2.7. Specifying an IP address**



The following points must be taken into consideration when specifying a DNS server manually. They depend on the intended subsequent use of the UCS system.

- When installing the first UCS system in a new UCS domain, the IP address of the local router (if it provides the DNS service) or the DNS server of the Internet provider should be entered.
- When installing every additional UCS system, the IP address of a UCS domain controller system must be specified as the DNS server. This is essential for the automatic detection of the domain controller master to function. In case of doubt, the IP address of the UCS domain controller master system should be entered.
- If the UCS system is to join a Windows Active Directory domain during the installation, the IP address of an Active Directory domain controller system should be specified as the DNS server. This is essential for the automatic detection of the Windows Active Directory domain controller to function.

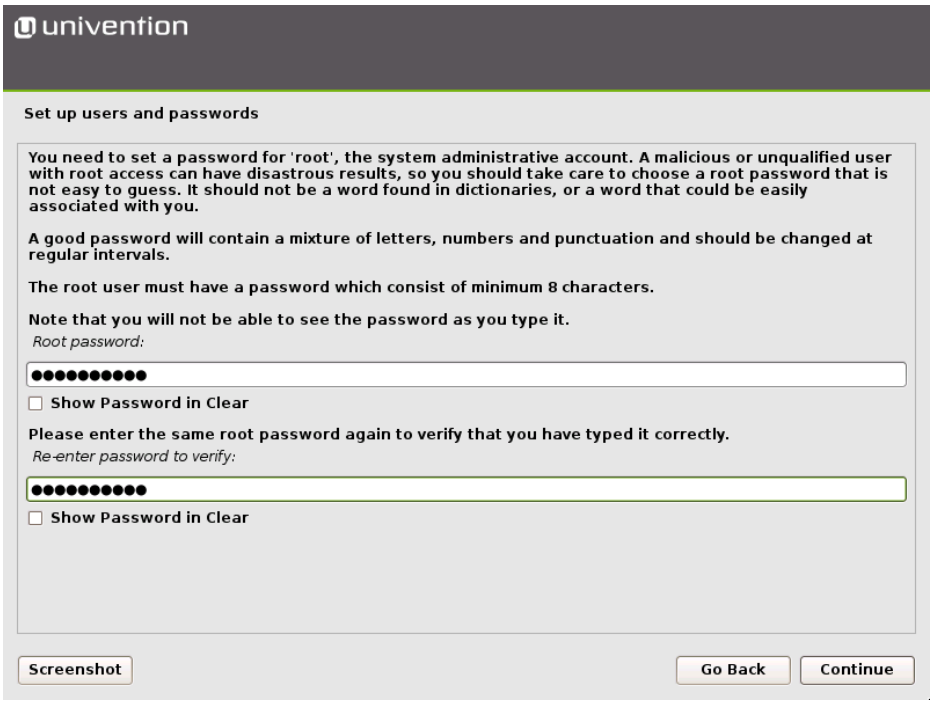
## 2.7. Setting up the root password

Feedback

Setting of a password for the `root` user is required for logging on to the installed system. If a master domain controller is installed, this password is also employed for the `administrator` user. In later operation, the passwords for the `root` and `administrator` users can be managed independently of each other. The password must be re-entered in the second entry field.

The password must contain at least eight characters for security reasons.

**Figure 2.8. Setting the root password**




## 2.8. Partitioning the hard drive

Feedback

The Univention Installer supports the partitioning of hard drives and the creation of different file systems (e.g., `ext4` and `XFS`). In addition, it is also possible to set up mechanisms such as the logical volume manager (LVM), RAID or partitions encrypted with LUKS.

As of UCS 4.0, the Univention Installer selects a suitable partition model (MBR or GPT) automatically depending on the size of the selected hard drive. On systems with the *Unified Extensible Firmware Interface (UEFI)*, the GUID Partition Table (GPT) is used automatically.

The Univention Installer offers guided installations to make installation simpler. In the guided installation, certain standard schemes with respect to the partitioning and formatting are applied to the selected hard drive. In addition, it is also possible to perform partitioning manually.

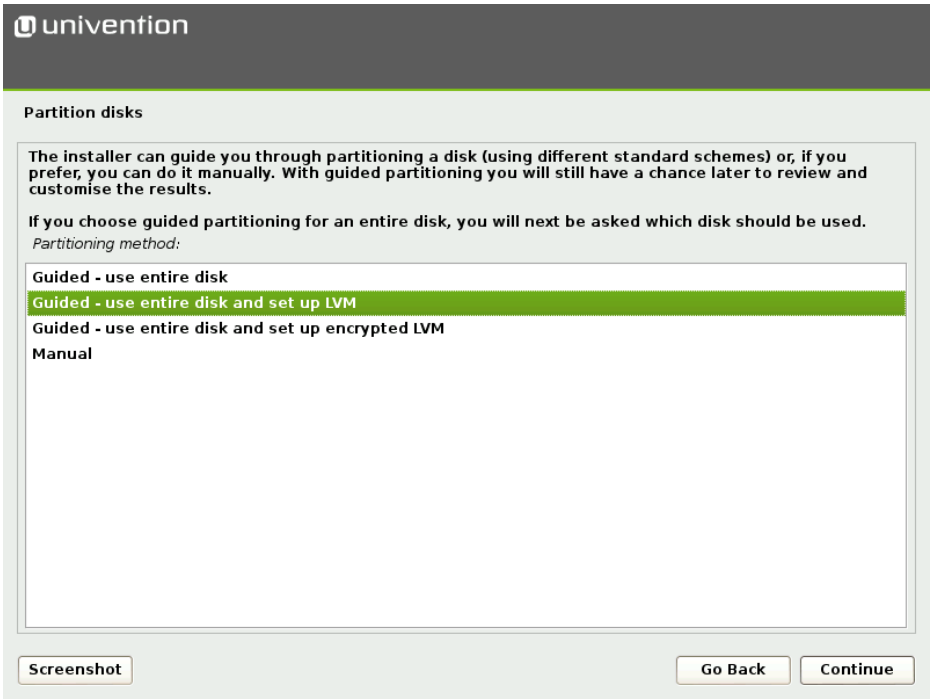
There are three schemes available for selection for guided partitioning:

- **Guided - Use entire disk**: In this scheme, an individual partition is created for each file system. Abstraction layers like LVM are not used. During the following step the number of filesystems/partitions is assigned. The size of the partitions is restricted to the size of the respective hard drive.
- **Guided - Use entire disk and set up LVM**: If the second scheme is selected, an LVM volume group is set up on the selected hard drive first. A separate logical volume is then created within the volume group for each file system. In this scheme, the size of the logical volume is restricted by the size of the volume group, which can also be subsequently enlarged with additional hard drives. In case of doubt, select this partitioning scheme.
- **Guided - Use entire disk with encrypted LVM**: This version is the same as the previous version, with the addition that the LVM volume group is also encrypted. Consequently, the password for the encrypted volume group has to be entered every time the system is started up.

## Caution

In all three versions, the data already on the selected hard drive are deleted during the partitioning!

**Figure 2.9. Selecting the partitioning scheme**



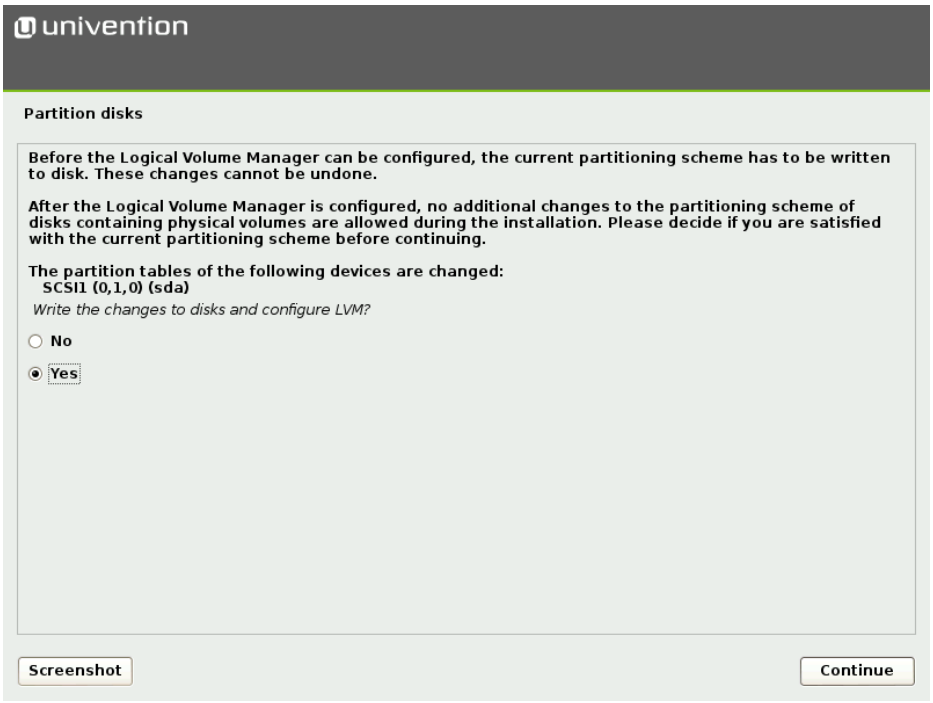

The next step is to select a hard drive from the list of those detected to which the partitioning version should be applied.

There are three subversions for each partitioning version, which differ in the number of file systems created:

- **All files in one partition**: In this version, just one partition or logical volume is created and the `/` file system saved there.
- **Separate /home partition**: In addition to a file system for `/`, an additional file system is also created for `/home/`.
- **Separate /home, /usr, /var and /tmp partition**: In addition to a file system for `/`, an additional file system is also created each for `/home/`, `/usr/`, `/var/` and `/tmp/`.

Before every active change to the hard drive, the change is displayed again in an additional dialogue and must be confirmed explicitly.

**Figure 2.10. Confirmation of changes to the hard drive**




Once the partitioning is complete, the UCS basic system and additional software is installed automatically. This can take some time depending on the speed of the hardware used. The system is then made ready to boot via the installation of the GRUB bootloader.

# 2.9. Domain settings

Feedback

The final configuration of the UCS system is started by selecting a domain mode. There are four modes available, which influence the following configuration steps:

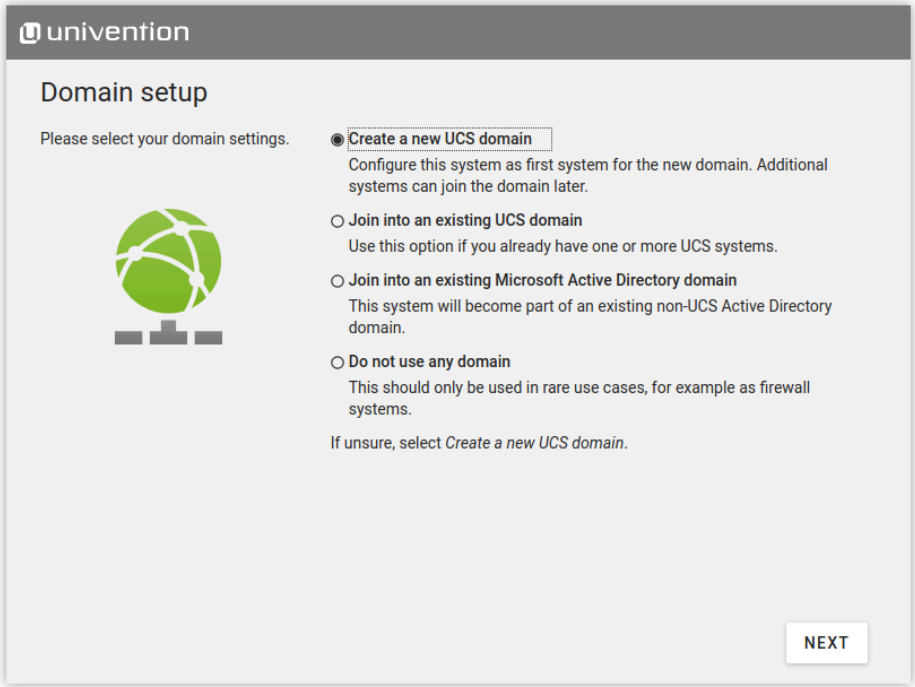
- In the first mode, **Create a new UCS domain**, the first system in a new UCS domain is configured: a UCS system with the *master domain controller* system role. In the following configuration steps, the information required for setting up a new directory service, authentication service and DNS server are requested. A UCS domain can comprise one single or several UCS systems. Additional UCS systems can be added at a later point in time using the **Join an existing UCS domain** mode.
- **Join into an existing Active Directory domain**: This mode, in which UCS is operated as a member of an Active Directory domain, is suitable for expanding an Active Directory domain with applications available

on the UCS platform. Apps installed on the UCS platform are then available for the users of the Active Directory domain to use. On selection of this mode, all the relevant information for the joining of the Active Directory domain is requested and the UCS system configured correspondingly.

- Selecting the **Join into an existing UCS domain** mode allows the UCS system to be configured to join an existing UCS domain. What UCS system role it is to take on in the domain is queried at a later stage.

- If the **Do not use any domain** mode is selected, there are no web-based administration functions and no domain functions at all available on the system. The UCS system can also not subsequently become a member of an existing UCS or Active Directory domain or found a new UCS domain at a later point in time. In addition, the Univention App Center is not available in this mode. For this reason, this mode is only used rarely and in special scenarios (e.g, as a firewall system).

**Figure 2.11. Domain settings**




## 2.9.1. "Create a new UCS domain" mode

Feedback

Once the **Create a new UCS domain** mode has been selected, an *organization name*, an *e-mail address*, a *fully qualified domain name* and an *LDAP base* are requested in the following two steps.

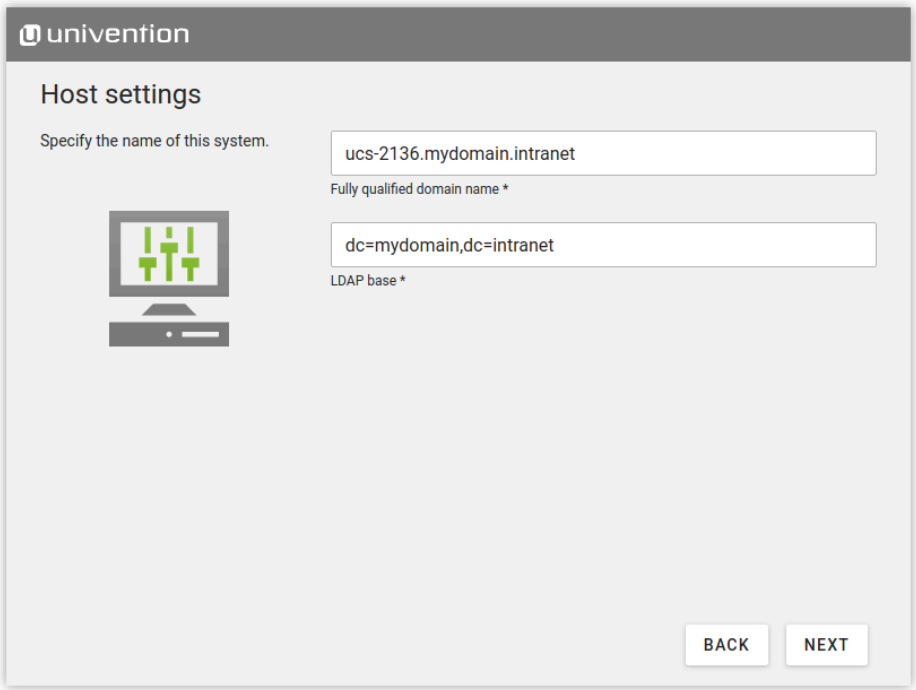
Specification of an organization name is optional and it is used in the second step to generate a domain name and the LDAP base automatically.

If a valid e-mail address is specified, this is used to activate a personalized license, which is required for the use of the Univention App Center. The license is generated automatically and sent to the specified e-mail address immediately. The license can then be imported via the Univention Management Console license dialog.

The name of the UCS system to be configured and the name of the DNS domain are determined from the fully qualified domain name (host name including domain name) entered here. A suggestion is generated automatically from the organization name entered in the previous step. It is recommended not to use a publicly available DNS domain, as this can result in problems during the name resolution.

A LDAP base needs to be specified for the initialization of the directory service. A suggestion is also derived here automatically from the fully qualified domain name. This value can usually be adopted without any changes.

**Figure 2.12. Specification of host name and LDAP base**




## 2.9.2. "Join an existing Active Directory domain" mode

Feedback

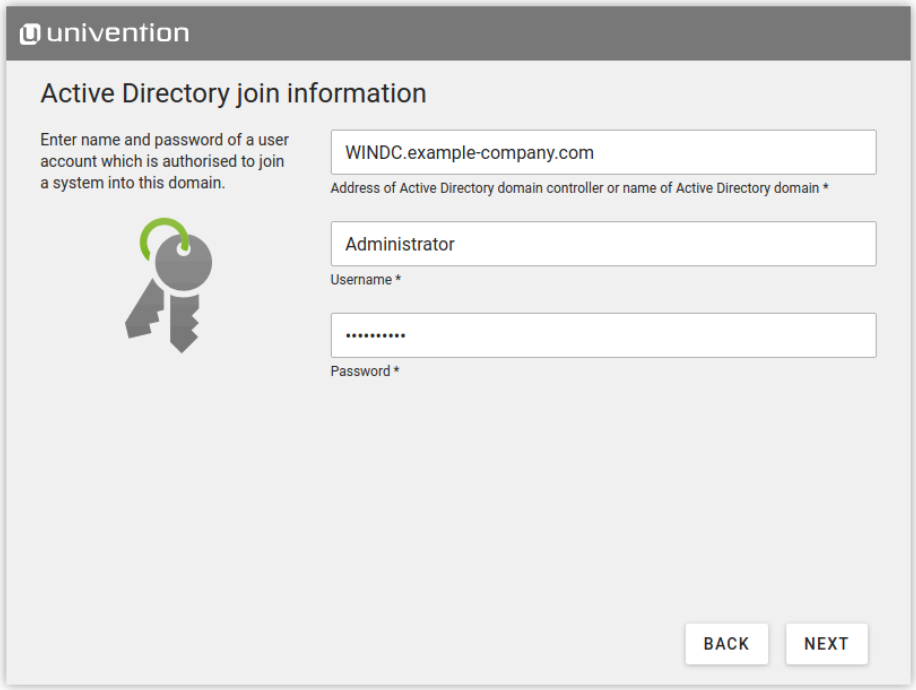
If the DNS server of an Active Directory domain was specified during the network configuration, the name of the Active Directory domain controller is suggested automatically in the **Active Directory account information** step. If the suggestion is incorrect, the name of another Active Directory domain controller or another Active Directory domain can be entered here.

The specification of an Active Directory account and the corresponding password is required for joining the Active Directory domain. The user account must possess the right to join new systems in the Active Directory domain.

In addition, a host name must be entered for the UCS system to be configured. The suggested host name can be adopted or a new host name entered. The domain name of the computer is derived automatically from the domain DNS server. In some scenarios (e.g., a public mail server) it can prove necessary to use a specific fully qualified domain name. The UCS system will join the Active Directory domain with the host name specified here. Once set up, the domain name *cannot* be changed again once the configuration is completed.

In a UCS domain, systems can be installed in different *system roles*. The first UCS system, that joins an Active Directory domain, is automatically installed with the master domain controller system role. If this mode is selected during installation of additional UCS systems, the system role selection dialogue is shown. The system roles are described within the following section.

**Figure 2.13. Information on the Active Directory domain join**




## 2.9.3. "Join an existing UCS domain domain" mode

Feedback

In a UCS domain, systems can be installed in different *system roles*. The first system in a UCS domain is always installed with the master domain controller system role. Additional UCS systems can join the domain at a later point in time and can be configured with one of the following system roles.

- **backup domain controller**

  The backup domain controller is the fallback system for the DC master. If the latter should fail, a DC backup can adopt the role of the DC master permanently. All the domain data and SSL security certificates are saved as read-only copies on servers with the backup domain controller role.

- **slave domain controller**

  All the domain data are saved as read-only copies on servers with the slave domain controller role. In contrast to the backup domain controller, however, not all security certificates are saved. As accesses to the services running on a slave domain controller are performed against the local LDAP directory service, DC slave systems are ideal for site servers and the distribution of high-load services.

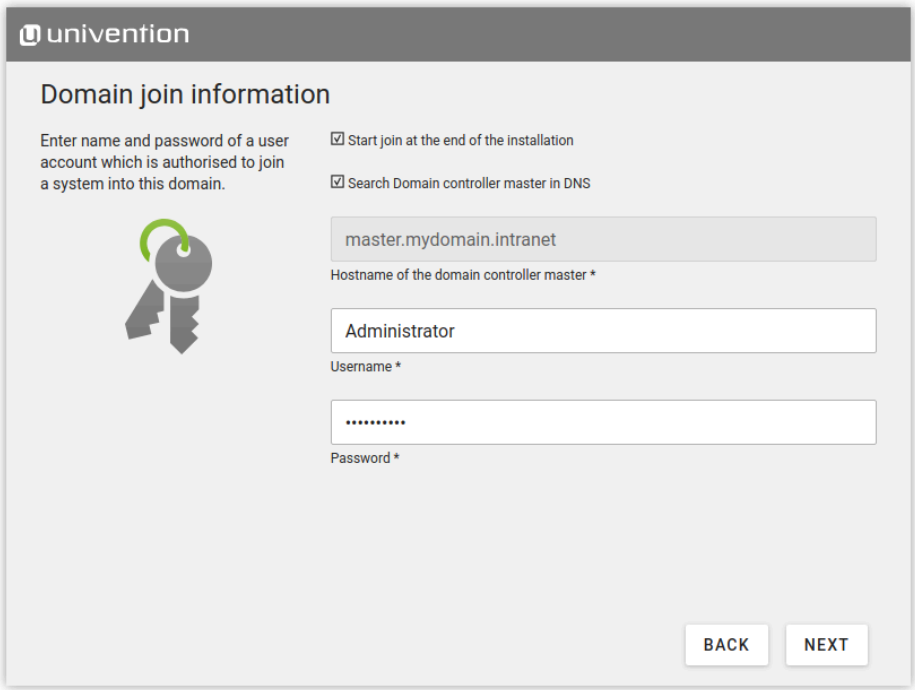
- **member server**

  member server are server systems without a local LDAP directory service. Access to domain data here is performed via other servers in the domain. They are therefore suitable for services which do not require a local database for authentication, for example, such as print and file servers.

Once the UCS system role has been selected, further information on the domain join is requested. If the domain join is not intended to occur automatically during the installation, the **Start join at the end of the installation** option can be disabled. If the correct DNS server was selected during the network configuration, Univention Installer can determine the name of the master domain controller system automatically. If the decision is taken to join another UCS domain, the **Search Domain controller master in DNS** option can

be disabled and the fully qualified domain name of the preferred master domain controller entered in the input field below. The access information required for the domain join must be entered in the **Administrator account** and **Administrator password** input fields.

**Figure 2.14. Information on the domain join**




In addition, a host name must be entered for the UCS system to be configured in the next step. The suggested host name can be adopted or a new host name entered. The domain name of the computer is derived automatically from the domain DNS server. In some scenarios (e.g., a public mail server) it can prove necessary to use a certain fully qualified domain name. Once set up, the domain name *cannot* be changed again once the configuration is completed.

### 2.9.4. "Do not use any domain" mode

Feedback

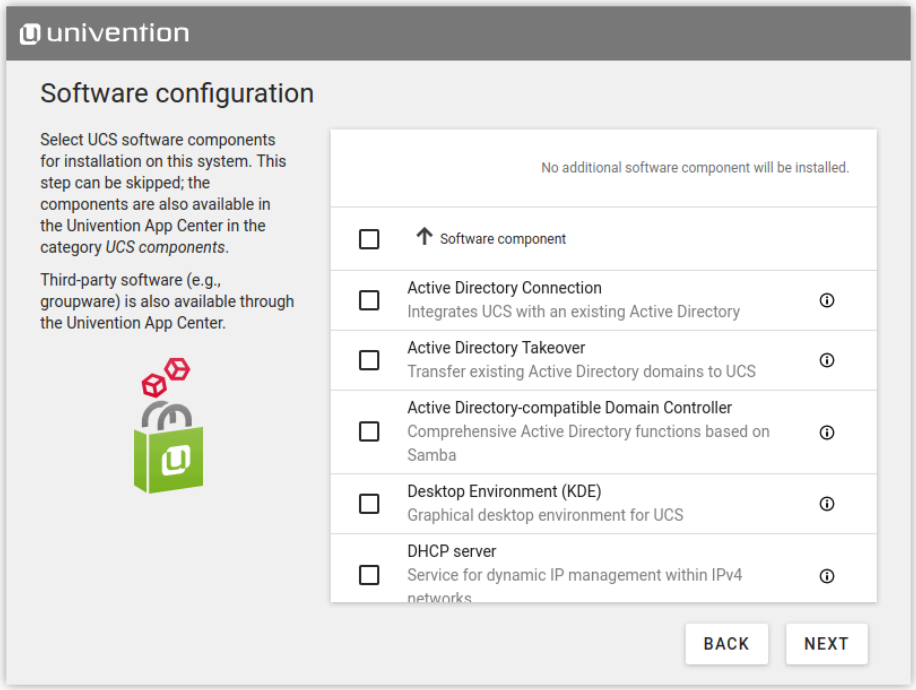
The configuration of the **Do not use any domain** mode requires the specification of a host name for the UCS system to be configured. The suggested host name can be adopted or a new host name entered. The domain name of the computer is derived automatically from the domain DNS server.

## 2.10. Selecting UCS software components

Feedback

The **software configuration** step offers the possibility of installing additional UCS components during the installation. The applications are also available after the installation via the Univention App Center in the **UCS components** category and can be installed and uninstalled there subsequently.

**Figure 2.15. Selecting UCS software components**



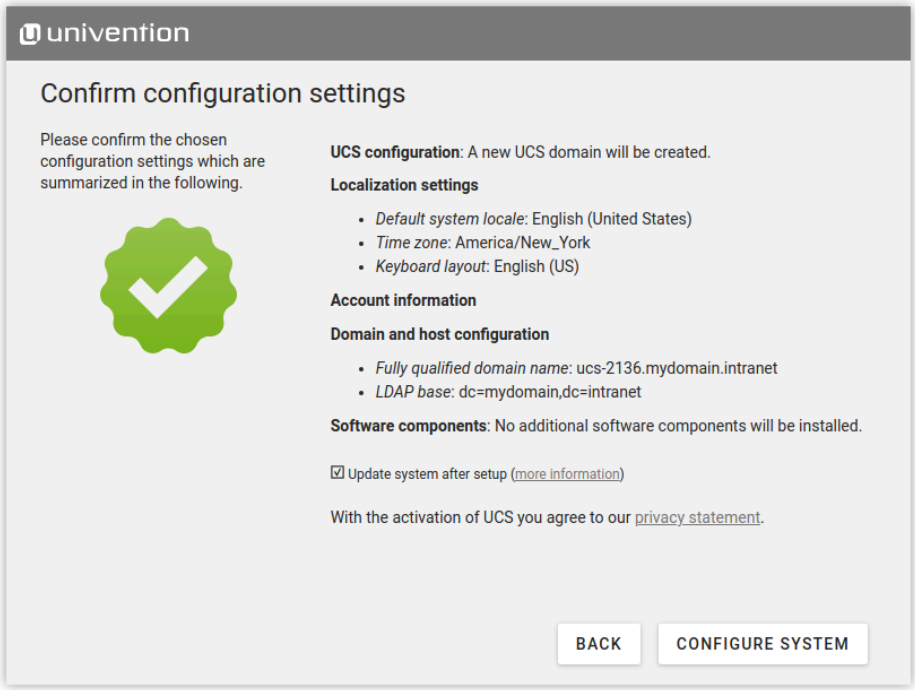

# 2.11. Confirming the settings

Feedback

This dialogue shows the major settings that were made. If all the settings are correct, the **Configure system** button can be used to start the configuration of the UCS system, see Figure 2.16.

The **Update system after installation** option allows the automatic installation of available Errata updates. In addition, all patch level updates and Errata updates available are installed on a master domain controller. On all other system roles, all the patch level updates are set up to the installation status of the master domain controller. (You need to log on to the master domain controller to check the installation status. This is done using the login data specified in the join options).

**Figure 2.16. Installation overview**




During the configuration, a progress bar displays the progress of the installation.

The installation protocol of the Univention Installer is saved in the following files:

- `/var/log/installer/syslog`
- `/var/log/univention/management-console-module-setup.log`

Completion of the configuration must be confirmed with the **Finish** button. The UCS system is then prepared for the first booting procedure and restarted.

The system will then boot from the hard drive. Following the boot procedure, the `root` and `administrator` users can log on via the web frontend Univention Management Console (see Chapter 4), which can be reached under the IP address set during the installation or the host name.

If the computer was installed as the first system in the UCS domain (master domain controller), the license can now be imported (see Section 4.4.2).

## 2.12. Troubleshooting for installation problems

Feedback

Information on possible installation problems can be found in the Univention Support database at http://sdb.univention.de in the subitem *Installation*.

## 2.13. Installation in text mode

Feedback

On systems that showed a problem with the graphic variant of Univention Installer, the installation may be also started in text mode. To achieve this, in the DVD boot menu **Advanced options** the entry **Install in text mode** has to be selected.

During installation in text mode Univention Installer shows the same information and asks for the same settings. After partitioning the hard drive, the system is prepared for the first boot and finally restarted.

After restart the configuration may be resumed by using a web browser. The URL `https://SERVER-IP-ADDRESS` or `http://SERVER-IP-ADDRESS` has to be opened within the browser (HTTPS is recommended). After loading the URI a login as user `root` is required.

The configuration process asks for location and network setting and then resumes with the same steps as the graphic variant of the installation, i.e. section *domain settings*.

## 2.14. Installation in the Amazon EC2 cloud

Feedback

Univention provides an Amazon Machine Image (AMI) for the Amazon EC2 cloud for UCS. This generic image for all UCS system roles is used to derive an individual instance which can be configured via the Univention Management Console (domain name, software selection, etc.).

The process for setting up a UCS instance based on Amazon EC2 is documented in the Univention Wiki [ec2-quickstart].

## 2.15. Installation in VMware

Feedback

If UCS is installed as a guest in VMware, the **Linux -> Other Linux system** option must be selected as the **Guest operating system** (UCS is based on Debian but the templates for Debian cannot be used).

The Linux kernel used in UCS includes all the support drivers necessary for operation in VMware (`vmw_balloon`, `vmw_pvsci`, `vmw_vmci`, `vmwgfx` and `vmxnet3`).

The open source version of the VMware Tools (Open VM Tools) is delivered with UCS. The tools can be installed using the ***open-vm-tools*** package (they are not required but do, for example, allow synchronization of the time on the virtualization server with the guest system).

## 2.16. Installation as Docker image

Feedback

Univention provides UCS as Docker images in the Docker Hub https://hub.docker.com/r/univention/. The description of the Docker images explains how they can be configured.

In a standard installation, the Docker images are used in a network that cannot be reached directly from outside the server. If multiple Docker images are used and they are executed on different Docker servers, a Software Defined Network or a VPN solution should be used.

## 2.17. Installation in Citrix XenServer

Feedback

The process for setting up a UCS instance in Citrix XenServer is documented in the Univention Wiki [xenserver-installation].

To display the GRUB menu correctly, an adaption to the XenServer configuration is necessary; this is described in [release-notes].

univention
be open.

# Chapter 3. Domain services / LDAP directory


3.1. Introduction ..... 38
3.2. Joining domains ..... 38
3.2.1. How UCS systems join domains ..... 38
3.2.1.1. Subsequent domain joins with `univention-join` ..... 39
3.2.1.2. Joining domains with Univention Management Console ..... 39
3.2.1.3. Join scripts / Unjoin scripts ..... 39
3.2.2. Windows domain joins ..... 40
3.2.2.1. Windows 10 ..... 41
3.2.2.2. Windows 8 ..... 41
3.2.2.3. Windows 7 ..... 42
3.2.2.4. Windows Server 2012 ..... 42
3.2.3. Ubuntu domain joins ..... 42
3.2.4. Mac OS X domain joins ..... 42
3.2.4.1. Domain join using the system preferences GUI ..... 42
3.2.4.2. Domain join on the command line ..... 43
3.3. UCS system roles ..... 43
3.3.1. Domain controller master ..... 43
3.3.2. Domain controller backup ..... 44
3.3.3. Domain controller slave ..... 44
3.3.4. Member server ..... 44
3.3.5. Base system ..... 44
3.3.6. Ubuntu ..... 44
3.3.7. Linux ..... 44
3.3.8. Univention Corporate Client ..... 44
3.3.9. Mac OS X ..... 44
3.3.10. Domain Trust Account ..... 45
3.3.11. IP managed client ..... 45
3.3.12. Windows Domaincontroller ..... 45
3.3.13. Windows Workstation/Server ..... 45
3.4. LDAP directory ..... 45
3.4.1. LDAP schemas ..... 45
3.4.1.1. LDAP schema extensions ..... 45
3.4.1.2. LDAP schema replication ..... 45
3.4.2. Audit-proof logging of LDAP changes ..... 46
3.4.3. Timeout for inactive LDAP connections ..... 47
3.4.4. LDAP command line tools ..... 47
3.4.5. Access control for the LDAP directory ..... 47
3.4.5.1. Delegation of the privilege to reset user passwords ..... 48
3.4.6. Name Service Switch / LDAP NSS module ..... 49
3.4.7. Syncrepl for synchronization with non-UCS OpenLDAP servers ..... 49
3.4.8. Configuration of the directory service when using Samba 4 ..... 49
3.4.9. Daily backup of LDAP data ..... 49
3.5. Listener/notifier domain replication ..... 49
3.5.1. Listener/notifier replication workflow ..... 49
3.5.2. Analysis of listener/notifier problems ..... 51
3.5.2.1. Log files/debug level of replication ..... 51
3.5.2.2. Identification of replication problems ..... 51
3.5.2.3. Reinitialization of listener modules ..... 51
3.6. SSL certificate management ..... 51

3.7. Kerberos ..... 53
3.8. Password hashes in the directory service ..... 53
3.9. SAML identity provider ..... 54
3.9.1. Login via *single sign-on* ..... 55
3.9.2. Adding a new external service provider ..... 55
3.9.3. Extended Configuration ..... 57
3.10. OpenID Connect Provider ..... 57
3.11. Converting a backup domain controller backup to the new master domain controller ..... 59
3.12. Fault-tolerant domain setup ..... 60
3.13. Protocol of activities in the domain ..... 61

# 3.1. Introduction

Feedback

Univention Corporate Server offers a cross platform domain concept with a common trust context between Linux and/or Windows systems. Within this domain a user is known to all systems via his username and password stored in the UCS management system and can use all services which are authorized for him. The management system keeps the account synchronized for the windows log-in, Linux/POSIX systems and Kerberos. The management of user accounts is described in Chapter 6.

All UCS and Windows systems within a UCS domain have a host domain account. This allows system-to-system authentication. Domain joining is described in Section 3.2.

The certificate authority (CA) of the UCS domain is operated on the master domain controller. A SSL certificate is generated there for every system that has joined the domain. Further information can be found in Section 3.6.

Every computer system which is a member of a UCS domain has a system role. This system role represents different permissions and restrictions, which are described in Section 3.3.

All domain-wide settings are stored in a directory service on the basis of OpenLDAP. Section 3.4 describes how to expand the managed attributes with LDAP scheme expansions, how to set up an audit-compliant LDAP documentation system and how to define access permissions to the LDAP directory.

Replication of the directory data within a UCS domain occurs via the Univention Directory Listener/Notifier mechanism. Further information can be found in Section 3.5.

Kerberos is an authentication framework the purpose of which is to permit secure identification in the potentially insecure connections of decentralized networks. Every UCS domain operates its own Kerberos trust context (realm). Further information can be found in Section 3.7.

# 3.2. Joining domains

Feedback

A UCS, Ubuntu or Windows system must join the domain after installation. The following describes the different possibilities to do this:

In addition to UCS, Ubuntu and Mac OS X, arbitrary Unix systems can be integrated into the domain. This is documented in [ext-doc-domain].

## 3.2.1. How UCS systems join domains

Feedback

There are three possibilities for a UCS system to join an existing domain; directly after installation in the Univention Installer (see Section 2.9.3) or subsequently using the command `univention-join` or using Univention Management Console.

The master domain controller should always be installed at the most up-to-date release stand of the domains, as problems can arise with an outdated domain control master when a system using the current version joins.

When a computer joins, a computer account is created, the SSL certificates are synchronized and an LDAP copy is initiated if necessary. The *join scripts* are also run at the end of the join process. These register further objects, etc., in the directory service using the software packages installed on the system (see Section 3.2.1.3).

The joining of the domain is registered on the client side in the `/var/log/univention/join.log` log file, which can be used for reference in error analysis. Actions run on the domain controller master are stored in the `/home/Join-Account/.univention-server-join.log` log file.

The joining process can be repeated at any time. Systems may even be required to rejoin following certain administrative steps (such as changes to important system features on the master domain controller).

### 3.2.1.1. Subsequent domain joins with `univention-join`

Feedback

`univention-join` retrieves a number of essential parameters interactively; however, it can also be configured using a number of parameters:

- The master domain controller is usually detected via a DNS request. If that is not possible (e.g., a DC slave server with a different DNS domain is set to join), the computer name of the DC master can also be entered directly using the `-dcname` *`HOSTNAME`* parameter. The computer name must then be entered as a fully qualified name, e.g., `master.company.com`.
- A user account which is authorized to add systems to the UCS domains is called a join account. By default, this is the `Administrator` user or a member of the two groups `Domain Admins` and `DC Backup Hosts`. The join account can be assigned using the `-dcaccount` *`ACCOUNTNAME`* parameter.
- The password can be set using the `-dcpwd` *`FILE`* parameter. The password is then read out of the specified file.
- The `-verbose` parameter is used to add additional debug output to the log files, which simplify the analysis in case of errors.

### 3.2.1.2. Joining domains with Univention Management Console

Feedback

A domain join can also be carried out web based via the UMC module **Domain join**. As the Administrator user does not yet exist on a system which has yet to join the domain, the login to Univention Management Console is done as user `root`.

As for the domain joining procedure via the command line, username and password of a user account authorized to add computers to a domain must be entered in the resulting dialogue. Likewise, the master domain controller will be determined automatically via a DNS request, but can also be entered manually.

The **Rejoin** option can be used to repeat the domain join at any time.

### 3.2.1.3. Join scripts / Unjoin scripts

Feedback

*Join scripts* are run during the domain join. Examples for changes made by join scripts are the registration of a print server in the domain or the adaptation of DNS entries. Join scripts are components of the individual software packages. In the same way, there are also *unjoin scripts*, which can reset these changes following uninstallation of software components.

Join scripts are stored in the `/usr/lib/univention-install/` directory and unjoin scripts in `/usr/lib/univention-uninstall/`. Each join/unjoin script has a version. An example: A package has al-

ready been installed and the join script already run. The new version of the package now requires additional changes and the version number of the join script is increased.

The `univention-check-join-status` command can be used to check whether join/unjoin scripts need to be run (either because they have yet to be run or an older version was run).

#### 3.2.1.3.1. Subsequent running of join scripts

Feedback

If there are join/unjoin scripts on a system which have not yet been run or which can only be run for an older version, a warning message is shown upon login to Univention Management Console.

Join scripts that have not been run can be executed via the UMC module **Domain join** by clicking on the menu entry **Execute all pending join scripts**.

The `univention-run-join-scripts` command is used to run all of the join/unjoin scripts installed on a system. The scripts check automatically whether they have already been executed.

The name of the join/unjoin script and the output of the script are also recorded in `/var/log/univention/join.log`.

If `univention-run-join-scripts` is run on another system role than the master domain controller, the user will be asked to input a username and password. This can be performed on the master domain controller via the `--ask-pass` option.

## 3.2.2. Windows domain joins

Feedback

The procedure for joining a Windows system to a UCS domain made available via Samba is now described as an example for Windows 7/8/10 and Windows 2012. The process is similar for other Windows versions. In addition to the client versions, Windows server systems can also join the domain. Windows servers join the domain as member servers; joining a Windows systems as a domain controller is not supported. Further information can be found in Section 9.1

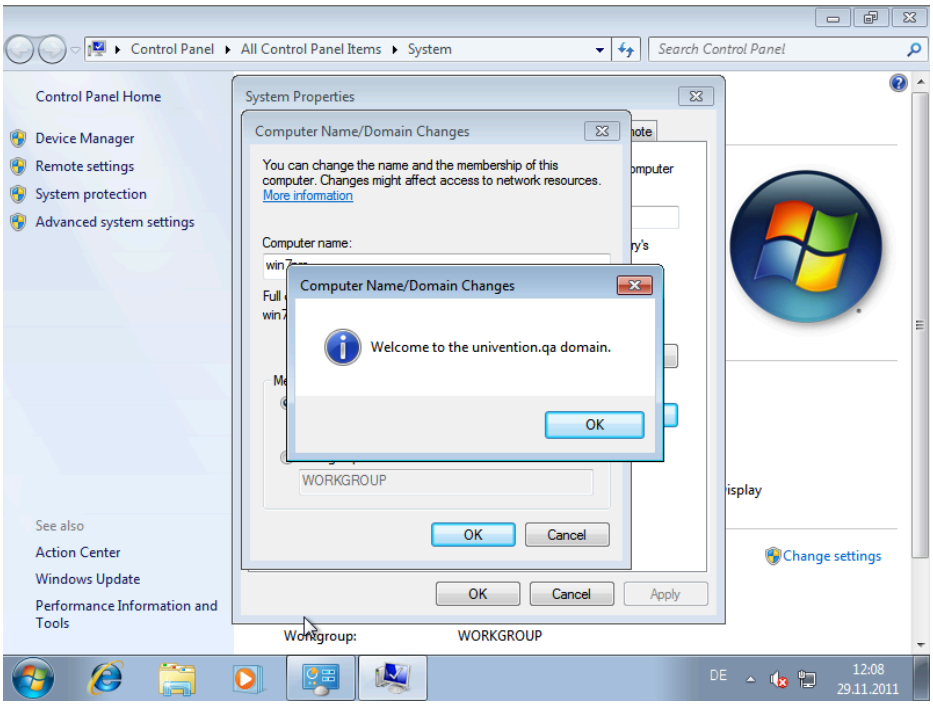
Only domain-compatible Windows versions can join the UCS domain, i.e., it is not possible for the Home versions of Windows to join a domain.

A host account is created for the Windows client automatically when it joins the domain (see Section 8.1). Information concerning MAC and IP addresses, the network, DHCP or DNS can be configured in Univention Management Console prior to or after joining the domain.

Domain joining is usually performed with the local Administrator account on the Windows system.

Joining the domain takes some time and the process must not be canceled prematurely. After successful joining a small window appears with the message **Welcome to the domain *`domain name`***. This should be confirmed with **[OK]**. The computer must then be restarted for the changes to take effect.

**Figure 3.1. Domain join of a Windows 7 system**




Domain names must be limited to 13 characters as they are otherwise truncated at the Windows client and this can lead to log-in errors.

For a domain join against a domain controller based on Samba 4, the DNS configuration of the client must be set up in such a way that DNS entries from the DNS zone of the UCS domain can also be resolved. In addition, the time on the client system must also be synchronized with the time on the domain controller.

### 3.2.2.1. Windows 10

Feedback

The joining of domains is only possible with the Pro and Enterprise editions of Windows 10.

The control panel can be reached via the search field **Search the web and Windows**, which can be found in the start bar. Under **System and Security -> System** it must be clicked on **Change settings -> Change**.

The **Domain** option field must be ticked and the name of the domain must be entered in the input field for the domain join. The full domain name should be used, e.g. **mydomain.intranet**. After clicking on the **[OK]** button, the username of a domain administrator must be entered in the input field **Username**, by default this is `Administrator`. The password of the domain administrator has to be entered in the input field **Password**. Finally, the process for joining the domain can then be started by clicking on **[OK]**.

### 3.2.2.2. Windows 8

Feedback

The joining of domains is only possible with the Pro and Enterprise editions of Windows 8.

The control panel can be reached by moving the cursor to the bottom right-hand corner of the screen. The *Control Panel* can then be searched for under **Search -> Apps**. **Change settings -> Network ID** must be clicked on under **System and Security -> System**.

The **Domain** option field must be ticked and the name of the Samba domain entered in the input field for the domain join. After clicking on the **[OK]** button, the `Administrator` must be entered in the input field **Name** and the password from `uid=Administrator,cn=users,`*`base DN`* transferred to the **Password** input field. The process for joining the domain can then be started by clicking on **[OK]**.

#### 3.2.2.3. Windows 7

Feedback

The joining of domains is only possible with the Professional, Enterprise or Ultimate editions of Windows 7.

The basic configuration dialogue is found under **Start -> Control Panel -> System and Security -> See the name of this computer**. **Change settings** must be selected and **Change** clicked under **Computer name, domain, and workgroup settings**.

The **Domain** option field must be ticked and the name of the Samba domain entered in the input field for the domain join. After clicking on the **[OK]** button, the `Administrator` must be entered in the input field **Name** and the password from `uid=Administrator,cn=users,`*`base DN`* transferred to the **Password** input field. The process for joining the domain can then be started by clicking on **[OK]**.

#### 3.2.2.4. Windows Server 2012

Feedback

The control panel can be reached by moving the cursor to the bottom right-hand corner of the screen. The *Control Panel* can then be searched for under **Search -> Apps**. **Change settings -> Network ID** must be clicked on under **System and Security -> System**.

The **Domain** option field must be ticked and the name of the Samba domain entered in the input field for the domain join. After clicking on the **[OK]** button, the `Administrator` must be entered in the input field **Name** and the password from `uid=Administrator,cn=users,`*`base DN`* transferred to the **Password** input field. The process for joining the domain can then be started by clicking on **[OK]**.

### 3.2.3. Ubuntu domain joins

Feedback

Univention provides the *Univention Domain Join Assistant* to integrate Ubuntu clients into a UCS domain. Documentation and installation instructions are available at Github[1].

### 3.2.4. Mac OS X domain joins

Feedback

UCS supports domain joins of Mac OS X clients into a UCS environment using Samba 4. This documentation refers to Mac OS X 10.8.2.

The domain join can be performed using the system preferences menu or the `dsconfigad` command line tool.

After the domain join it is possible to automatically mount CIFS shares to subfolders in `/Volumes` when logging in with a domain user. For that, the following line has to be added to the file `/etc/auto_master`:

```
/Volumes    auto_custom
```

In addition, the file `/etc/auto_custom` needs to be created and the shares which should be mounted have to be listed in it in the following way:

```
subfolder name-fstype=smbfs    ://fqdn/sharename
```

Note that the automounted shares are not displayed in the finder's sidebar.

#### 3.2.4.1. Domain join using the system preferences GUI

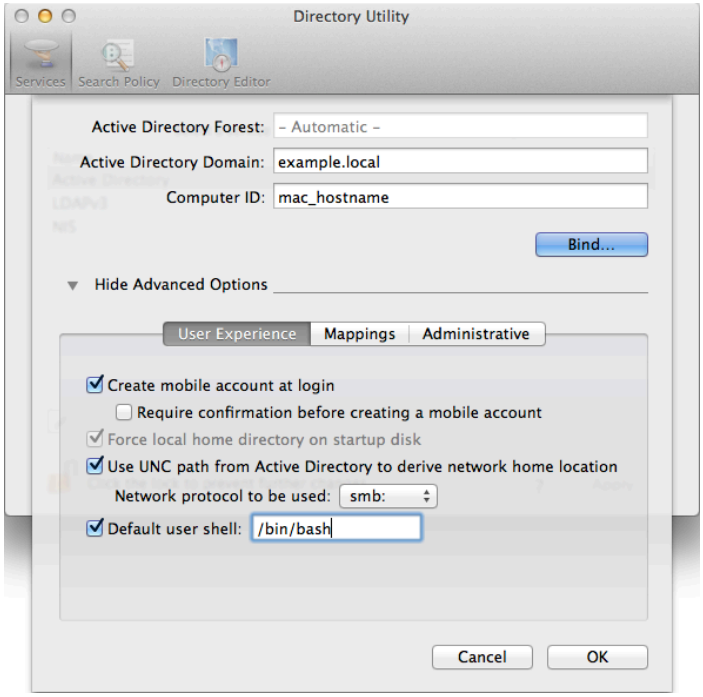
Feedback

In the System Preferences via the **Users & Groups** entry, the **Login menu** can be reached. After authenticating by clicking on the lock in the lower left corner and providing credentials of a local Administrator account,

[1] https://github.com/univention/univention-domain-join

the **Network Account Server: Join** button needs to be clicked. From that menu it is possible to open the **Directory Utility**.

**Figure 3.2. Domain join of a Mac OS X system**




In the advanced options section, the option **Create mobile account at login** should be activated. A mobile account has the advantage that, when the domain is not available, the user can log into the Mac OS X system with the same account used for logging into the domain.

After filling in the domain name in the field **Active Directory Domain** and the hostname of the Mac OS X client in the field **Computer ID**, the join process is initiated after clicking the button **Bind...**. The username and password of an account in the `Domain Admins` group needs to be entered, e.g., `Administrator`.

### 3.2.4.2. Domain join on the command line

Feedback

The domain join can also be performed on the command line using `dsconfigad`:

```
dsconfigad -amac hostname-domainfqdn-ou "CN=Computers,ldap_base" \
  -uDomain Administrator-mobile enable
```

Additional configuration options are available through `dsconfigad -help`.

## 3.3. UCS system roles

Feedback

In a UCS domain systems can be installed in different *system roles*. The following gives a short characterization of the different systems:

### 3.3.1. Domain controller master

Feedback

A system with the master domain controller role (DC master for short) is the primary domain controller of a UCS domain and is always installed as the first system. The domain data (such as users, groups, printers) and the SSL security certificates are saved on the DC master.

Copies of these data are automatically transferred to all servers with the backup domain controller role.

### 3.3.2. Domain controller backup

Feedback

All the domain data and SSL security certificates are saved as read-only copies on servers with the backup domain controller role (backup DC for short).

The backup domain controller is the fallback system for the master domain controller. If the latter should fail, a backup domain controller can take over the role of the master domain controller permanently (see Section 3.11).

### 3.3.3. Domain controller slave

Feedback

All the domain data are saved as read-only copies on servers with the slave domain controller role (slave DC for short). In contrast to the backup domain controller, however, not all security certificates are synchronized.

As access to the services running on a slave domain controller are performed against the local LDAP server, slave DC systems are ideal for site servers and the distribution of load-intensive services.

A slave DC system cannot be promoted to a master DC.

### 3.3.4. Member server

Feedback

member server are server systems without a local LDAP server. Access to domain data here is performed via other servers in the domain.

### 3.3.5. Base system

Feedback

A base system is an autonomous system which is not a member of the domain. It is not connected to any LDAP server. It still provides the UCS update mechanism and Univention Configuration Registry for configuration, but not the graphical administration interface Univention Management Console.

A basic system is thus suitable for services which are operated outside of the trust context of the domain, such as a web server or a firewall.

### 3.3.6. Ubuntu

Feedback

Ubuntu clients can be managed with this system role, see Section 8.1.1.

### 3.3.7. Linux

Feedback

This system role is used for the integration of other Linux systems than UCS and Ubuntu, e.g., for Debian or CentOS systems. The integration is documented in [ext-doc-domain].

### 3.3.8. Univention Corporate Client

Feedback

A Univention Corporate Client is a desktop or thin client system based on Univention Corporate Client.

### 3.3.9. Mac OS X

Feedback

Mac OS X systems can be joined into a UCS domain using Samba 4. Additional information can be found in Section 3.2.4.

### 3.3.10. Domain Trust Account

Feedback

A domain trust account is set up for trust relationships between Windows and UCS domains.

### 3.3.11. IP managed client

Feedback

An IP managed client allows the integration of non-UCS systems into the IP management (DNS/DHCP), e.g., for network printers or routers.

### 3.3.12. Windows Domaincontroller

Feedback

Windows domain controllers in a Samba 4 environment are operated with this system role.

### 3.3.13. Windows Workstation/Server

Feedback

Windows clients and Windows member servers are managed with this system role.

## 3.4. LDAP directory

Feedback

Univention Corporate Server saves domain-wide data in a LDAP directory service based on OpenLDAP. This chapter describes the advanced configuration and coordination of OpenLDAP.

Often several LDAP servers are operated in a UCS domain. The configuration of the server(s) used is described in Section 8.4.5.

### 3.4.1. LDAP schemas

Feedback

Schema definitions specify which object classes exist and which attributes they include, i.e., which data can be stored in a directory service. Schema definitions are saved as text files and included in the OpenLDAP server's configuration file.

UCS uses standard schemas where possible in order to allow interoperability with other LDAP applications. Schema extensions are supplied for Univention-specific attributes - such as for the policy mechanism.

#### 3.4.1.1. LDAP schema extensions

Feedback

To keep the efforts required for small extensions in LDAP as low as possible, Univention Corporate Server provides its own LDAP scheme for customer extensions. The LDAP object class `univentionFreeAttributes` can be used for extended attributes without restrictions. It offers 20 freely usable attributes (*univentionFreeAttribute1* to *univentionFreeAttribute20*) and can be used in connection with any LDAP object (e.g., a user object).

If LDAP schema extensions are to be delivered as part of software packages, there is also the possibility of packaging them and distributing them to all the backup domain controller servers in the domain using a Univention Directory listener module. Further information is available in [packaging-schema-extensions].

#### 3.4.1.2. LDAP schema replication

Feedback

The replication of the LDAP schemas is also automated via the listener/notifier mechanism (see Section 3.5). This relieves the administrator of the need to perform all schema updates manually on all the OpenLDAP servers in the domain. Performing the schema replication before the replication of LDAP objects guarantees that this doesn't fail as a result of missing object classes or attributes.

On the master domain controller, a checksum for all the directories with schema definitions is performed when the OpenLDAP server is started. This checksum is compared with the last saved checksum in the `/var/lib/univention-ldap/schema/md5` file.

The actual replication of the schema definitions is initiated by the Univention Directory Listener. Prior to every request from the Univention Directory Notifier for a new transaction ID, its current schema ID is requested. If this is higher than the schema ID on the listener side, the currently used sub-schema is procured from the notifier system's LDAP server via an LDAP search.

The output sub-schema is included on the listener system in LDIF format in the `/var/lib/univention-ldap/schema.conf` file and the local OpenLDAP server restarted. If the schema replication is completed with this step, the replication of the LDAP objects is continued.

## 3.4.2. Audit-proof logging of LDAP changes

Feedback

The ***univention-directory-logger*** package allows the logging of all changes in the LDAP directory service. As each data record contains the hash value of the previous data record, manipulations of the log file - such as deleted entries - can be uncovered.

Individual areas of the directory service can be excluded from the logging. These branches can be configured using the Univention Configuration Registry variables `ldap/logging/exclude1`, `ldap/logging/exclude2`, etc. As standard, the container is excluded in which the temporary objects are stored (`cn=temporary,cn=univention`). The LDAP changes are logged by a Univention directory listener module. The Univention directory listener service must be restarted if changes are made to the Univention Configuration Registry variables.

The logging is made in the `/var/log/univention/directory-logger.log` file in the following format:

```
START
Old Hash: Hash sum of the previous data record
DN: DN of the LDAP object
ID: Listener/notifier transaction ID
Modifier: DN of the modifying account
Timestamp: Time stamp in format dd.mm.yyyy hh:mm:ss
Action: add, modify or delete

Old Values:
 List of old attributes, empty when an object is added
New Values:
 List of new attributes, empty when an object is deleted
END
```

A hash sum is calculated for each logged data record and also logged in the *daemon.info* section of the Syslog service.

As of UCS 4.4-0 erratum 536[1] the respective transaction ID of the entry is added to the file `/var/log/univention/directory-logger.log` before each line as a prefix:

```
ID 342: START
ID 342: Old Hash: 70069d51a7e2e168d7c7defd19349985
ID 342: DN: uid=Administrator,cn=users,dc=example,dc=com
ID 342: ID: 342
ID 342: Modifier: cn=admin,dc=example,dc=com
```

[1] https://errata.software-univention.de/#/?erratum=4.4x536

```
ID 342: Timestamp: 15.04.2020 09:20:40
ID 342: Action: modify
ID 342:
ID 342: Old values:
ID 342: description: Dhis is a description test
ID 342: entryCSN: 20200415091936.317108Z#000000#000#000000
ID 342: modifyTimestamp: 20200415091936Z
ID 342:
ID 342: New values:
ID 342: description: This is a description test
ID 342: entryCSN: 20200415092040.430976Z#000000#000#000000
ID 342: modifyTimestamp: 20200415092040Z
ID 342: END
```

If ***univention-directory-logger*** was installed before this UCS version, the old behavior (no prefix) is retained by default. By setting the Univention Configuration Registry variable `ldap/logging/id-prefix=yes` the new behavior can be activated. This prefix simplifies the correlation of related lines when post-processing the log in analysis and monitoring software.

## 3.4.3. Timeout for inactive LDAP connections

Feedback

The Univention Configuration Registry variable `ldap/idletimeout` is used to configure a time period in seconds after which the LDAP connection is cut off on the server side. When the value is set to 0, no expiry period is in use. The timeout period has been set at six minutes as standard.

## 3.4.4. LDAP command line tools

Feedback

In addition to the UMC web interface, there are also a range of programs with which one can access the LDAP directory from the command line.

The `univention-ldapsearch` tool simplifies the authenticated search in the LDAP directory. A search filter needs to be specified as an argument; in the following example, the administrator is searched for using the user ID:

```
univention-ldapsearch uid=Administrator
```

The `slapcat` command makes it possible to save the current LDAP data in a text file in LDIF format, e.g.:

```
slapcat > ldapdata.txt
```

## 3.4.5. Access control for the LDAP directory

Feedback

Access to the information contained in the LDAP directory is controlled by Access Control Lists (ACLs) on the server side. The ACLs are defined in the central configuration file `/etc/ldap/slapd.conf` and managed using Univention Configuration Registry. The `slapd.conf` is managed using a multifile template; further ACL elements can be added below `/etc/univention/templates/files/etc/ldap/slapd.conf.d/` between the `60univention-ldap-server_acl-master` and `70univention-ldap-server_acl-master-end` files or the existing templates expanded upon.

If LDAP ACL extensions are to be delivered as part of software packages, there is also the possibility of packaging them and distributing them to all the LDAP servers in the domain using a Univention Directory listener module. Further information is available in [packaging-acl-extensions].

The default setting of the LDAP server after new installations with UCS does not allow anonymous access to the LDAP directory. This behavior is configured with the Univention Configuration Registry variable `ldap/`

`acl/read/anonymous`. Individual IP addresses can be granted anonymous read permissions via Univention Configuration Registry variable `ldap/acl/read/ips`.

Following successful authentication on the LDAP server, all attributes of a user account can be read out by this user.

In addition, an extra, internal account, the root DN, also has full write access.

In addition, UCS offers a number of further ACLs installed as standard which suppress access to sensitive files (e.g., the user password) and establish rules which are necessary for operation (e.g., necessary accesses to computer accounts for log-ins). The read and write access to this sensitive information if only intended for members of the `Domain Admins` group. Nested groups are also supported. The Univention Configuration Registry variable `ldap/acl/nestedgroups` can be used to deactivate the nested groups function for LDAP ACLs, which will result in a speed increase for directory requests.

### 3.4.5.1. Delegation of the privilege to reset user passwords

Feedback

To facilitate the delegation of the privilege to reset user passwords, the ***univention-admingrp-user-passwordreset*** package can be installed. It uses a join script to create the `User Password Admins` user group, in so far as this does not already exist.

Members of this group receive the permission via additional LDAP ACLs to reset the passwords of other users. These LDAP ACLs are activated automatically during the package installation. To use another group, or a group that already exists, instead of the `User Password Admins` group, the DN of the group to be used can be entered in the Univention Configuration Registry variable `ldap/acl/user/passwordreset/accesslist/groups/dn`. The LDAP server must be restarted after making changes.

Passwords can be reset via Univention Management Console. In the default setting, Univention Management Console only offers the user wizard to the `Administrator` user, which allows the setting of new passwords. During the installation a new `default-user-password-admins` policy is created automatically, which is linked to the members of the `User Password Admins` group and can be assigned to a corresponding container in the LDAP directory. Further information on the configuration of UMC policies can be found in Section Section 4.9.

The policy makes it possible to search for users and create an overview of all the attributes of a user object. If an attempt is made to modify further attributes in addition to the password when the user does not have sufficient access rights to the LDAP directory, Univention Directory Manager denies him write access with the message *Permission denied*.

#### Caution

The package should be installed on the domain controller master and the domain controller backup systems. During the installation, the LDAP server is restarted and is thus temporarily unavailable.

Password resets via the password group can be prevented for sensitive users or groups (e.g., domain administrators). The Univention Configuration Registry variables `ldap/acl/user/passwordreset/protected/uid` and `ldap/acl/user/passwordreset/protected/gid` can be used to configure users and groups. Multiple values must be separated by commas. After changes to the variables, it is necessary to restart the LDAP server using the `/etc/init.d/slapd restart` command. In the default setting, the members of the `Domain Admins` group are protected against having theirs password changed.

If access to additional LDAP attributes should be necessary for changing the password, the attribute names can be expanded in Univention Configuration Registry variable `ldap/acl/user/passwordreset/attributes`. After the change, the LDAP directory service must be restarted for the change to take effect. This variable is already set appropriately for a UCS standard installation.

### 3.4.6. Name Service Switch / LDAP NSS module

Feedback

With the *Name Service Switch*, the GNU C standard library (`glibc`) used in Univention Corporate Server offers a modular interface for resolving the names of users, groups and hosts.

The LDAP NSS module is used on UCS systems for access to the domain data (e.g., users) as standard. The module queries the LDAP server specified in the Univention Configuration Registry variable `ldap/server/name` (and if necessary the `ldap/server/addition`).

What measures should be taken if the LDAP server cannot be reached can be specified by the Univention Configuration Registry variable `nssldap/bindpolicy`. As standard, if the server cannot be reached, a new connection attempt is made. If the variable is set to `soft`, then no new attempt is made to connect. This can considerably accelerate the boot of a system if the LDAP server cannot be reached, e.g., in an isolated test environment.

### 3.4.7. Syncrepl for synchronization with non-UCS OpenLDAP servers

Feedback

The syncrepl replication service can also be activated parallel to the notifier service for the synchronization of OpenLDAP servers not installed on UCS systems. Syncrepl is a component of OpenLDAP, monitors changes in the local directory service and transmits them to other OpenLDAP servers.

### 3.4.8. Configuration of the directory service when using Samba 4

Feedback

As standard, the OpenLDAP server is configured in such a way that it also accepts requests from ports 7389 and 7636 in addition to the standard ports 389 and 636.

If Samba 4 is used, the Samba domain controller service occupies the ports 389 and 636. In this case, OpenLDAP is automatically reconfigured so that only ports 7389 and 7636 are used. This must be taken into account during the configuration of syncrepl in particular (see Section 3.4.7). `univention-ldapsearch` uses the standard port automatically.

### 3.4.9. Daily backup of LDAP data

Feedback

The content of the LDAP directory is backed up daily on the master domain controller and all backup domain controller systems via a Cron job. If Samba 4 is used, its data directory is also backed up.

The LDAP data are stored in the `/var/univention-backup/` directory in the naming scheme `ldap-backup_DATE.ldif.gz` in LDIF format. They can only be read by the `root` user. The Samba 4 files are stored in the directory `/var/univention-backup/samba/`.

The Univention Configuration Registry variable `backup/clean/max_age` can be used to define how long old backup files are kept (e.g. `backup/clean/max_age=365`, all files older than 365 days are automatically deleted). For new installations (from UCS 4.4-7 on) the default for this variable is 365 (days). If the variable is not set, no backup files are deleted.

## 3.5. Listener/notifier domain replication

Feedback

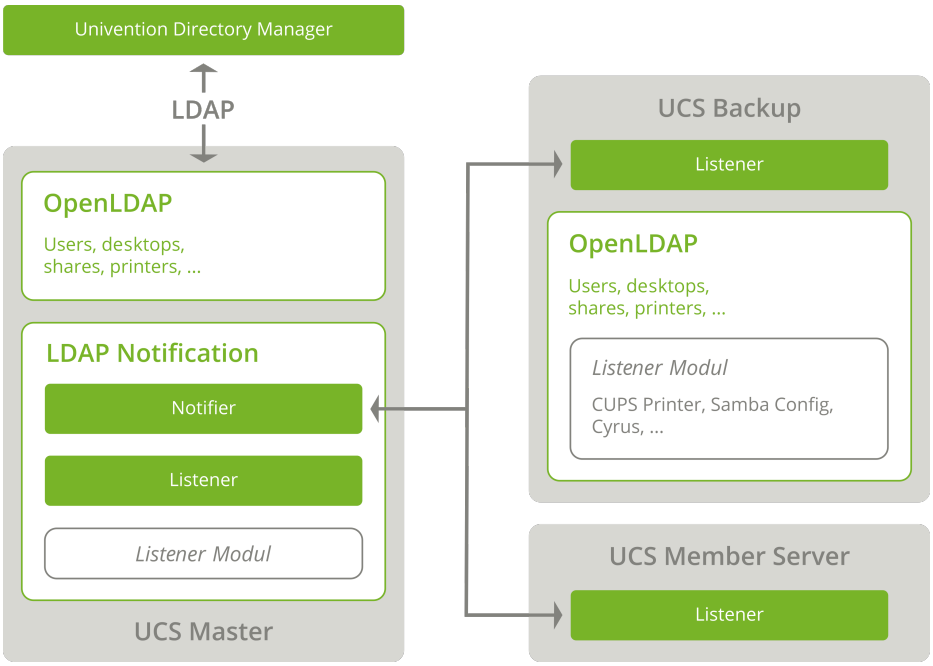
### 3.5.1. Listener/notifier replication workflow

Feedback

Replication of the directory data within a UCS domain occurs via the Univention Directory Listener/Notifier mechanism:

- The Univention Directory Listener service runs on all UCS systems.
- On the master domain controller (and possibly existing backup domain controller systems) the *Univention Directory Notifier* service monitors changes in the LDAP directory and makes the selected changes available to the Univention Directory Listener services on the other UCS systems.

**Figure 3.3. Listener/Notifier mechanism**




The active Univention Directory Listener instances in the domain connect to a Univention Directory Notifier service. If an LDAP change is performed on the master domain controller (all other LDAP servers in the domain are read-only), this is registered by the Univention Directory Notifier and notified to the listener instances.

Each Univention Directory Listener instance uses a range of Univention Directory Listener modules. These modules are shipped by the installed applications; the print server package includes, for example, listener modules which generate the CUPS configuration.

Univention Directory Listener modules can be used to communicate domain changes to services which are not LDAP-compatible. The print server CUPS is an example of this: The printer definitions are not read from the LDAP, but instead from the `/etc/cups/printers.conf` file. Now, if a printer is saved in the UMC printer management, it is stored in the LDAP directory. This change is detected by the Univention Directory Listener module *cups-printers* and an entry added to, modified or deleted in `/etc/cups/printers.conf` based on the data in the LDAP.

Additional information on the setup of Univention Directory Listener modules and developing your own modules can be found in [developer-reference].

LDAP replication is also performed by a listener module. If the LDAP server to be replicated to is not accessible, the LDAP changes are temporarily stored in the `/var/lib/univention-directory-replication/failed.ldif` file. The contents of the file are automatically transferred to the LDAP when the LDAP server is available again.

The listener/notifier mechanism works based on transactions. A transaction ID is increased for every change in the LDAP directory of the master domain controller. A Univention Directory Listener instance which has missed several transactions - for example, because the computer was switched off - automatically requests

all the missing transactions once the connection is available again until its local transaction ID corresponds to that of the master domain controller.

## 3.5.2. Analysis of listener/notifier problems

Feedback

### 3.5.2.1. Log files/debug level of replication

Feedback

All status messages from the Univention Directory Listener and the executed listener modules are logged in the `/var/log/univention/listener.log` file. The level of detail of the log messages can be configured using the Univention Configuration Registry variable `listener/debug/level`. The possible values are from 0 (only error messages) to 4 (all status messages). Once the debug level has been changed, the Univention Directory Listener must be restarted.

Status messages from the Univention Directory Notifier service are logged in the `/var/log/univention/notifier.log` file. The debug level can be configured using the `notifier/debug/level` variable (also from 0-4). Once the debug level has been changed, the Univention Directory Notifier must be restarted.

### 3.5.2.2. Identification of replication problems

Feedback

When the domain replication is running normally (normal system load, no network problems), the delay between the change being made in Univention Management Console and replicated to, for example, a slave domain controller is barely noticeable. An incomplete replication can be identified by comparing the transaction IDs of the listener and notifier services.

The transactions registered by the notifier service are written in the `/var/lib/univention-ldap/notify/transaction` file in ascending order on the master domain controller. An example:

```
root@dcmaster:~# tail -1 /var/lib/univention-ldap/notify/transaction
836 cn=dcslave3,cn=dc,cn=computers,dc=firma,dc=de m
```

The last transaction received by the listener system is stored in the `/var/lib/univention-directory-listener/notifier_id` file:

```
root@dcslave1:~# cat /var/lib/univention-directory-listener/notifier_id
836
```

This check can also be performed automatically by the Nagios service `UNIVENTION_REPLICATION` (see Section 15.2.2.1).

### 3.5.2.3. Reinitialization of listener modules

Feedback

If there are problems in running a listener module, there is the option of reinitializing the module. In this case, all LDAP objects with which the listener module works are passed on again.

The name of the listener module must be supplied to the command for the renewed initialization. The installed listener modules can be found in the `/var/lib/univention-directory-listener/handlers/` directory.

The following command can be used to reinitialize the printer module, for example:

```
univention-directory-listener-ctrl resync cups-printers
```

# 3.6. SSL certificate management

Feedback

In UCS, sensitive data are always sent across the network encrypted, e.g., via the use of SSH for the login to systems or via the use of protocols based on SSL/TLS. (*Transport Layer Security (TLS)* is the current protocol

name, the name of the previous protocol *Secure Socket Layer (SSL)*, however, is still more common and is also used in this documentation).

For example, SSL/TLS is employed in the listener/notifier domain replication or for HTTPS access to Univention Management Console.

Both communication partners must be able to verify the authenticity of the key used for encrypted communication between two computers. To this end, each computer also features a so-called *host certificate*, which is issued and signed by a certification authority (CA).

UCS provides its own CA, which is automatically set up during installation of the master domain controller and from which every UCS system automatically procures a certificate for itself and the CA's public certificate when joining the domain. This CA appears as the root CA, signs its own certificate and can sign certificates for other certification authorities.

The properties of the CA are generated automatically during the installation based on system settings such as the locale. These settings can be subsequently adapted on the master domain controller in the UMC module **Certificate settings**.

## Caution

If the UCS domain contains more than one system, all other host certificates need to be reissued after changing the root certificate! The procedure required for this is documented in SDB 1183[1].

The UCS-CA is always found on the master domain controller. A copy of the CA is stored on every backup domain controller, which is synchronized with the CA on the domain controller master by a Cron job every 20 minutes.

## Caution

The CA is synchronized from the master domain controller to the backup domain controller and not vice-versa. For this reason, only the CA on the master domain controller should be used.

If a backup domain controller is promoted to the master domain controller (see Section 3.11), the CA on the new master domain controller can be used directly.

The UCS root certificate has a specified validity period - as do the computer certificates created with it.

## Caution

Once this period of time elapses, services which encrypt their communication with SSL (e.g., LDAP or domain replication) no longer function.

It is thus necessary to verify the validity of the certificate regularly and to renew the root certificate in time. A Nagios plugin is provided for the monitoring of the validity period. In addition, a warning is shown when logging on to Univention Management Console if the root certificate is going to expire soon (the warning period can be specified with the Univention Configuration Registry variable `ssl/validity/warning`; the standard value is 30 days).

The renewal of the root certificate and the other host certificates is documented in SDB 1183[1].

On UCS systems, a Cron job verifies the validity of the local computer certificate and the root certificate daily and records the expiry date in the Univention Configuration Registry variables `ssl/validity/host`

[1] http://sdb.univention.de/1183
[1] http://sdb.univention.de/1183

(host certificate) and `ssl/validity/root` (root certificate). The values entered there reflect the number of days since the 1/1/1970.

In Univention Management Console, the effective expiry date of the computer and root certificate can be accessed via the upper right user menu and the entry **License -> License information**.

## 3.7. Kerberos

Feedback

Kerberos is an authentication framework the purpose of which is to permit secure identification in the potentially insecure connections of decentralized networks. In Kerberos, all clients use a foundation of mutual trust, the *Key Distribution Center* (KDC). A client authenticates at this KDC and receives an authentication token, the so-called ticket which can be used for authentication within the Kerberos environment (the so-called Kerberos realm). The name of the Kerberos realm is configured as part of the installation of the master domain controller and stored in the Univention Configuration Registry variable `kerberos/realm`. It is not possible to change the name of the Kerberos realm at a later point in time.

Tickets have a standard validity period of 8 hours; this is why it is vital for a Kerberos domain to have the system time synchronized for all the systems belonging to the Kerberos realm.

Univention Corporate Server uses the Heimdal Kerberos implementation. An independent Heimdal service is started on UCS domain controller systems without Samba 4, while Kerberos is provided by a Heimdal version integrated in Samba on Samba 4 DCs. In a environment composed of UCS domain controllers without Samba 4 and Samba 4 domain controllers both Kerberos environments are based on identical data (these are synchronized between Samba 4 and OpenLDAP via the Univention S4 connector (see Section 9.2.2.4)).

As standard, the KDC is selected via a DNS service record. The KDC used by a system can be reconfigured using the Univention Configuration Registry variable `kerberos/kdc`. If Samba 4 is installed on a system in the domain, the service record is reconfigured so that only the Samba 4-based KDCs are offered. In a mixed environment it is recommended only to use the Samba 4 KDCs.

The Kerberos admin server, on which the administrative settings of the domain can be made, runs on the master domain controller. Most of the settings in Univention Corporate Server are taken from the LDAP directory, so that the major remaining function is changing passwords. This can be achieved by means of the Tool `kpasswd`; the passwords are then changed in the LDAP too. The Kerberos admin server can be configured on a system via the Univention Configuration Registry variable `kerberos/adminserver`.

## 3.8. Password hashes in the directory service

Feedback

User password hashes are stored in the directory service in the *userPassword* attribute. The `crypt` library function is used to hash passwords. The actual hashing method can be configured via the Univention Configuration Registry variable `password/hashing/method`, `SHA-512` is used by default.

As an alternative Univention Corporate Server (from version UCS 4.4-7 erratum 887[1] on) offers the option of using `bcrypt` as hashing method for passwords of user accounts. To activate `bcrypt` support in OpenLDAP the Univention Configuration Registry variable `ldap/pw-bcrypt=true` has to bet set on all LDAP servers. Otherwise it is not possible authenticate with a `bcrypt` hash as password hash. Additionally the Univention Configuration Registry variable `password/hashing/bcrypt=true` has to be set, again on all servers, to activate `bcrypt` as the hashing method for setting or changing user password.

In addition, the `bcrypt` cost factor and the `bcrypt` variant can be configured via the Univention Configuration Registry variables `password/hashing/bcrypt/cost_factor` (default `12`) and `password/hashing /bcrypt/prefix` (default `2b`).

[1] https://errata.software-univention.de/#/?erratum=4.4x887

### Caution

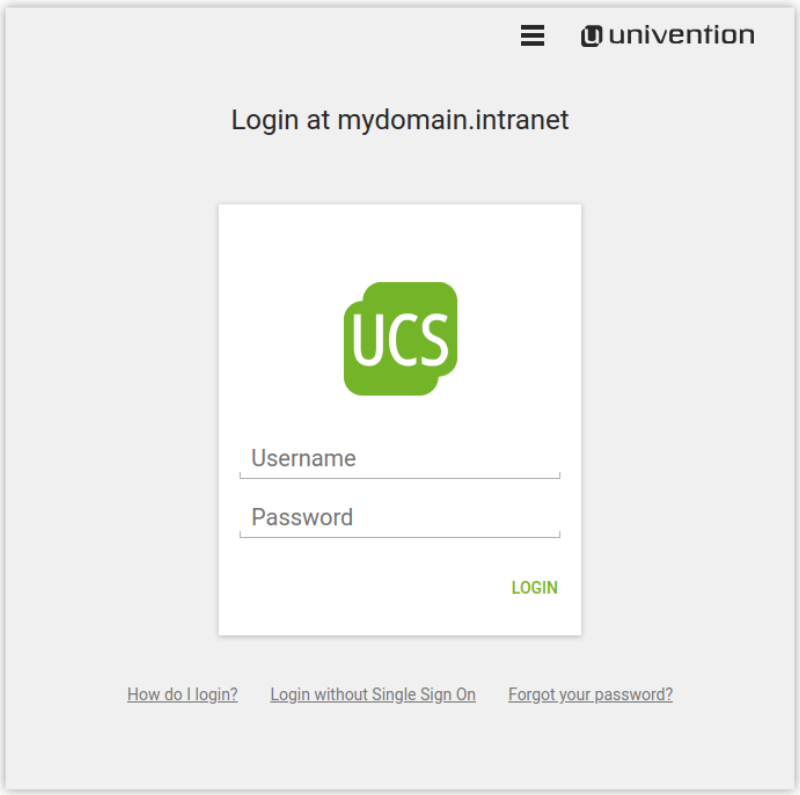
`bcrypt` is limited to a maximum of 72 characters. So only the first 72 characters of the password are used to generate the hashes.

## 3.9. SAML identity provider

Feedback

SAML (Security Assertion Markup Language) is an XML-based standard for exchanging authentication information in order to allow single sign-on across domain boundaries. UCS provides a fail-safe SAML identity provider on a master domain controller as well as backup domain controller. The SAML identity provider is registered at an external service with a cryptographic certificate and establishes a trust relationship. The user then only needs to authenticate himself against UCS and can use the service without renewed authentication.

**Figure 3.4. The *single sign-on* login page**




The SAML 2.0 compatible UCS identity provider is provided by the integration of ***simplesamlphp***.

The UCS identity provider is tightly integrated into the UCS domain. Clients that will be used to access the UCS identity provider have to be able to resolve DNS records in the UCS domain. The domain DNS Servers should therefore be configured on all clients in order to be able to resolve the central DNS record, which by default is `ucs-sso.`*`domainname`*.

The UCS identity provider is automatically installed on master domain controller and backup domain controllers. Further backup domain controllers can be made available in the domain to increase fail-safe safety. The default DNS record `ucs-sso.`*`domainname`* is registered to increase fail-safe access to the UCS identity provider. The SSL certificate for this record is kept on all participating systems in the domain. It is advised to install the UCS domain root certificate on all clients that are using *single sign-on*.

It is possible to associate the SAML authentication with the Kerberos login. This means that users with a valid Kerberos ticket, for example after logging on to Windows or Linux, can log on to the identity provider without having to manual re-authenticate.

To allow Kerberos authentication at the identity provider, the Univention Configuration Registry variable `saml/idp/authsource` has to be changed from `univention-ldap` to `univention-negotiate`. The web browsers must be configured to transfer the Kerberos ticket to the SAML Identity Provider. Here are two examples for the configuration of Firefox and Internet Explorer / Microsoft Edge:

Mozilla Firefox

In the extended Firefox configuration, which can be reached by entering **about:config** in the Firefox address line, the address of the identity provider must be entered in the option **network.negotiate-auth.trusted-uris**, which is `ucs-sso.`*`domainname`* by default.

Microsoft Internet Explorer,
Microsoft Edge

In the Control Panel, the **Internet Options** must be opened, followed by **Security**, **Local Intranet**, **Sites**, **Advanced**. The address of the identity provider has to be added, which is `ucs-sso.`*`domainname`* by default.

The Kerberos authentication can be restricted to certain IP subnets by setting the Univention Configuration Registry variable `saml/idp/negotiate/filter-subnets` for example to `127.0.0.0/16,192.168.0.0/16`. This is especially useful to prevent a pop up login box being shown for clients which are not part of the UCS domain.

## 3.9.1. Login via *single sign-on*

Feedback

The *single sign-on* is the default login for Univention Management Console, as long as `ucs-sso.`*`domainname`* can be reached. To login the domain credentials must be provided. For the login directly at the UCS system (i.e., without *single sign-on*), follow the link **Login without Single Sign On**.

The design of the login dialog can be changed by editing `/usr/share/univention-management-console-login/css/custom.css`. This file will never be altered or deleted during updates.

Other web services will redirect to the UCS identity provider login page in a similar fashion in order to carry out a *single sign-on*. After authenticating, the user will be forwarded back to the web service itself. These services need to be registered as described in Section 3.9.2.

The *single sign-on* for a particular service can be initiated from the UCS identity provider, as well. This saves an extra visit at the external web service which redirects to the authentication site. To do so, a link to the UCS identity provider page needs to be provided in the form of `https://ucs-sso.`*`domainname`*`/simplesamlphp/saml2/idp/SSOService.php?spentityid=`*`[Service provider identifier]`*.

## 3.9.2. Adding a new external service provider

Feedback

The Univention Management Console domain module **SAML identity provider** allows to manage all service providers that are registered at the UCS identity provider. Users have to be activated for a service provider, to be able to authenticate for it at the UCS identity provider. The service provider can be activated for groups, to allow authentication with that service provider for all users within that group. On the users **Account tab** or the groups **General tab**, the service provider can to be added under **SAML settings**.

To register the UCS identity provider at an external service provider, the public part of the SAML certificate is required by the service provider. The certificate can be downloaded via a link in the UMC module. Some service providers may require the UCS identity provider XML metadata as a file upload. In the default config-

uration, the XML file can be downloaded from the URL `https://ucs-sso.domainname/simplesamlphp/saml2/idp/metadata.php`.

The following attributes can be configured when adding a new service provider.

### Table 3.1. General options when configuring a service provider

| Attribute | Description |
|---|---|
| Service provider activation status | If activated, the configuration for the service provider is activated and is ready for authentication. |
| Service provider identifier | Defines the internal name of the service provider. The name is later selected at user objects, when giving them access to a service provider. The identifier cannot be changed later. |
| Respond to this service provider URL after login | After successful authentication, the user's browser is redirected to the service provider. The redirection is done to this provided URL. |
| Single logout URL for service provider | Service providers can offer a URL endpoint at which the session at the service provider can be terminated. If a user logs out at the UCS identity provider, the browser will get redirected to the provided URL to terminate the session. |
| Format of `NameID` attribute | The value `NameIDFormat` that the service provider receives. The service provider's documentation should contain information about possible values. Example: `urn:oasis:names:tc:SAML:2.0:nameid-format:transient` or `urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`. |
| Name of the attribute that is used as `NameID` | The LDAP attribute that is used to uniquely identify the user is provided here, e.g., `uid`. |
| Name of the organization for service provider | The value provided here will be shown on the UCS single sign-on login page. It helps the user to identify for which service he enters credentials. |
| Description of this service provider | The value provided here will be shown on the UCS single sign-on login page. A longer description about the service provider can be given here. The description will be shown on the login page in a separate paragraph. |

### Table 3.2. Advanced settings when configuring a service provider

| Attribute | Description |
|---|---|
| URL to the service provider's privacy policy | If a URL is entered here, the UCS identity provider login page will contain a link to this URL. |
| Allow transmission of LDAP attributes to the service provider | By default, the UCS identity provider transmits only the `NameID` attribute entered on the **General** page to the service provider. If additional LDAP user attributes are required by the service provider, this checkbox can be activated. The attributes that should be transmitted have to be entered in the **List of LDAP attributes to transmit**. |
| Value for `attribute format` field | In case the transmitted attributes need to be sent in a particular format value, this format can be entered here. Example: `urn:oasis:names:tc:SAML:2.0:nameid-format:transient` or `urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`. |
| List of LDAP attributes to transmit | Every LDAP attribute that should be transmitted to the service provider can be entered here. Additionally, one or more service attribute names |

| Attribute | Description |
|---|---|
| | can be added to each LDAP attribute in the field next to it. These service attribute names have the purpose to translate the LDAP Attribute names for the service provider. Multiple service attribute names have to be separated by commas. In order for the UCS identity provider to process these attributes, they need to be registered additionally via the LDAP object `id=default-saml-idp,cn=univention,`*`base DN`*. LDAP attributes entered at the object can be read and transferred by the Identity Provider. |

### 3.9.3. Extended Configuration

Feedback

Some environments may require the UCS Identity Provider to provide multiple logical Identity Provider instances. Logical separation is achieved by offering different URIs as Identity Provider endpoints. The default endpoint is `https://ucs-sso.`*`domainname`*`/simplesamlphp/saml2/idp/metadata.php`. Further entries can be created by setting Univention Configuration Registry variables in the form `saml/idp/entityID/supplement/`*`identifier`*`=true` on all servers which serve the UCS Identity Provider. Typically that will be the DC master and all DC backup servers. The `apache2` service must then be reloaded. For example, to set up another entry under the URI `https://ucs-sso.`*`domainname`*`/simplesamlphp/`*`secondIDP`*`/saml2/idp/metadata.php`, the Univention Configuration Registry variable `saml/idp/entityID/supplement/secondIDP=true` must be set.

## 3.10. OpenID Connect Provider

Feedback

UCS offers the possibility to install a OpenID Connect Provider, which allows external web services to delegate the user login via the OpenID Connect (OIDC) protocol to the UCS Identity Management. The OpenID Connect Provider App can be installed via the App Center. The service is provided by the software Kopano Konnect.

The app can be installed on all system roles. When installing on a UCS system with the role master domain controller or backup domain controller the OpenID Connect Provider is made available under the DNS entry for the *Single Sign-On*, normally this is `ucs-sso.`*`domain.name`*. If the app is installed on a different system role, the provider can be reached directly via the hostname instead. It should be ensured that the app is installed on all other servers that are reachable at the `ucs-sso` DNS CNAME. Session synchronization between multiple installed OIDC Providers in a domain is not preconfigured. When experiencing login issues with Apps, we recommend to only install the OIDC Provider on one system, and restrict the `ucs-sso` DNS CNAME to that system, or contact Univention Support.

External Web services can be connected to UCS via OpenID Connect by creating a specific object of type `oidc/rpservice` for this service in the UCS directory service. These can be created in the Univention Management Console in the LDAP browser in the container `cn=oidc`, which is located below the container `cn=univention`. Here the new service can be registered via the item Add and the selection **OpenID Connect Relying Party Service**.

The same is also possible from the command line:

```
udm oidc/rpservice create --set name=<UCS_internal_identifier> \
  --position="cn=oidc,cn=univention,$(ucr get ldap/base)" \
  --set clientid=<ClientID> \
  --set clientsecret=<a_long_password> \
  --set trusted=yes \
  --set applicationtype=web \
  --set redirectURI=<URL_from_service_documentation>
```

The command parameters are:

`name`

the service name displayed in the web interface during login.

`clientid`,
`secret`

must be identical here and in the connected service (*shared secret*).

`trusted`

should be set to `yes` by default. Otherwise, the user will be prompted for confirmation to transfer their user attributes to the service.

`applicationtype`

should be set to `web` for Internet services.

`redirectURI`

URL of the login endpoint, which can be found in the documentation of the connected service. If a service is accessible via several URLs or should it also be accessible via IP address, all possible addresses must be added to the `redirectURI` attribute. The field can therefore be defined multiple times, whereby each individual value must contain a valid URL.

The connected web service still needs information about the OpenID Connect endpoints of the provider app for its configuration. If the provider app is installed, this information can be found at the URL `https://ucs-sso.`*`domain.name`*`/.well-known/openid-configuration`. If the provider app was installed on a system other than master domain controller or backup domain controller, use the FQDN of the respective server instead of `ucs-sso.domain.name` as described above.

When using OpenID Connect, resolvable DNS names and verifiable certificates are a prerequisite. This is especially true for client computers of end users who need to access both the DNS resolvable host names of the Web service and the OpenID Connect Provider. In addition, the externally connected Web services must be able to establish a connection to the OpenID Connect Provider in order to be able to retrieve the user attributes.

In the special case where the DNS name of the OIDC provider is to be changed, the corresponding value must first be adjusted in the app settings of the OpenID Connect Provider App. Since there are diverse scenarios for the availability of the provider after changing the DNS name, the web server configuration cannot be changed automatically. For example, depending on the configured DNS name, the UCS Apache configuration has to be adapted. The configuration file `/etc/apache2/conf-available/openid-connect-provider.conf` must be made available under the set DNS name in a virtual host.

With version 2 of the OIDC-Provider App the authentication to OpenID Connect works via the SAML Identity Provider of the UCS domain. If the SAML Identity Provider is not reachable at the default URL `https://ucs-sso.`*`domain.name`*, the correct URL under which the SAML IdP metadata for the UCS domain can be retrieved must be entered correctly in the app settings. If this URL is configured incorrectly, the OpenID Connect Provider will not start.

With SAML authentication, the authorization for the use of the OpenID Connect Provider and thus for all apps connected via OIDC can be controlled via SAML authorizations. By default, the group `domain users` is enabled for access when the app is installed. If this permission should be removed, the corresponding option must also be activated in the app settings so that the permission is not automatically added again.

The OpenID Connect Provider logs actions via the Docker Daemon. The output can be viewed with the command `univention-app logs openid-connect-provider`.

# 3.11. Converting a backup domain controller backup to the new master domain controller

Feedback

A UCS domain consists of only one master domain controller, but is not limited in the number of backup domain controller. A backup domain controller stores all the domain data and all SSL security certificates as read-only copies. However, in contrast to the master domain controller, writing changes are not allowed.

Any backup domain controller can be converted to a master domain controller. There are two typical scenarios for this:

- In an emergency if the hardware of the master domain controller fails
- To replace a fully functional master domain controller with new hardware or changing the architecture from i386 to amd64.

### Caution

The conversion of a backup domain controller to a master domain controller is a serious configuration change and should be prepared carefully. The conversion cannot be reversed.

The master domain controller that is going to be replaced has to be shut down before the conversion. It must not be powered on during or after the conversion!

Before the conversion, the installed software packages and the current configuration has to be compared between the master domain controller and backup domain controller. If the master domain controller is not available anymore, use a file backup. After the conversion, all possibly remaining references of the old master domain controller have to be removed or changed to the new master domain controller.

The conversion primarily involves the changeover of the services relevant for authentication such as LDAP, DNS, Kerberos and Samba. The installed software needs to be adjusted manually (this can be done using the UMC modules **App Center** or **Package Management**). For example, if the mail component was installed on the previous master domain controller, it will not be automatically installed on the new master domain controller after the conversion. To minimize manual changes after the conversion, please consider Section 3.12.

If additional LDAP schema packages were installed on the master domain controller, they must also be installed on the backup domain controller prior to the conversion. The package list of the old master domain controller should be saved prior to the promotion in order to allow a subsequent comparison of the installed packages. The package list can be created with the following command:

```
dpkg --get-selections \* >> dpkg.selection
```

This file should be compared with the same output on the backup domain controller. Missing packages should then be installed on the backup domain controller. Especially those packages that install a LDAP schema are absolutely necessary. The following command executed on the master domain controller will list all affected packages:

```
ls -1 /etc/ldap/schema/*.schema /usr/share/univention-ldap/schema/
*.schema | xargs dpkg -S
```

To simply install all packages of the master domain controller also on the backup domain controller, use the previously created file `dpkg.selection` of the master domain controller and run the following command on the backup domain controller:

```
dpkg --set-selections < dpkg.selection
apt-get dselect-upgrade
```

In addition, the Univention Configuration Registry inventory needs to be saved so that it is possible to compare the configuration adjustments on the new master domain controller. The following files on the master domain controller need to be compared with those on the backup domain controller:

```
/etc/univention/base.conf
/etc/univention/base-forced.conf
```

UCS saves a copy of those files every night to `/var/univention-backup/ucr-backup_%Y%m%d.tgz`

The conversion of a backup domain controller to the new master domain controller is performed by running the command `/usr/lib/univention-ldap/univention-backup2master` on the backup domain controller. The system must be rebooted after the conversion. The process is logged to `/var/log/univention/backup2master.log`

The following steps are performed by `univention-backup2master`:

- Checking the environment: The system must be a backup domain controller that already joined the domain. Additionally, it is checked if the master domain controller can be resolved via DNS and if the repository server can be reached. Also, the master domain controller must be powered off and not reachable anymore.
- Now, the most important services OpenLDAP, Samba, Kerberos and Univention Directory Notifier and Listener will be stopped. Important Univention Configuration Registry variable, such as `ldap/master` and `server/role` will be changed. The UCS Root CA certificate will be available via the webserver on the backup domain controller. All mentioned services will be started again.
- The DNS SRV record `kerberos-adm` will be changed from the old to the new master domain controller.
- If present, the Univention S4 Connector (see Section 9.2.2.4) will be removed from the computer object of the old master domain controller and will be scheduled for re-configuration on the new master domain controller.
- The server role of the new master domain controller will be changed to `domaincontroller_master` in the OpenLDAP directory service. The DNS SRV record `_domaincontroller_master._tcp` will also be adjusted.
- If present, all entries of the old master domain controller will be removed from the local Samba directory service. Additionally, the FSMO roles will be transferred to the new master domain controller.
- The computer object of the old master domain controller will be deleted from OpenLDAP.
- The OpenLDAP directory service will be searched for any remaining references to the old master domain controller. All found references (e.g. DNS records) are shown and suggested to be fixed. The suggested fixes have to be checked and confirmed one by one.
- Finally, the package ***univention-server-backup*** will be replaced by ***univention-server-master***.

Subsequently, the LDAP directory on the new master domain controller and the Univention Configuration Registry values on all UCS systems of the domain should be checked for any remaining references to the hostname or the IP address of the old master domain controller. Those references need to be adjusted to the new master domain controller, too.

## 3.12. Fault-tolerant domain setup

Feedback

In a domain exist some services that are important for the functionality of all of its members. Redundancy can be used to remove those single points of failure. An article in the Univention Support database explains how to secure LDAP, Kerberos, DNS, DHCP and Active Directory-compatible Domain Controllers: SDB 1349[1].

[1] http://sdb.univention.de/1349

# 3.13. Protocol of activities in the domain

Feedback

The *Admin Diary* app provides the facility to log important events happening in the domain. This includes among others:

- Creation, move, modification and deletion of users and other objects using Univention Directory Manager
- Installation, update and deinstallation of apps
- Server password changes
- Start, end and eventual failures of domain joins
- Start and end of UCS updates

**Figure 3.5. View of events in min Diary**

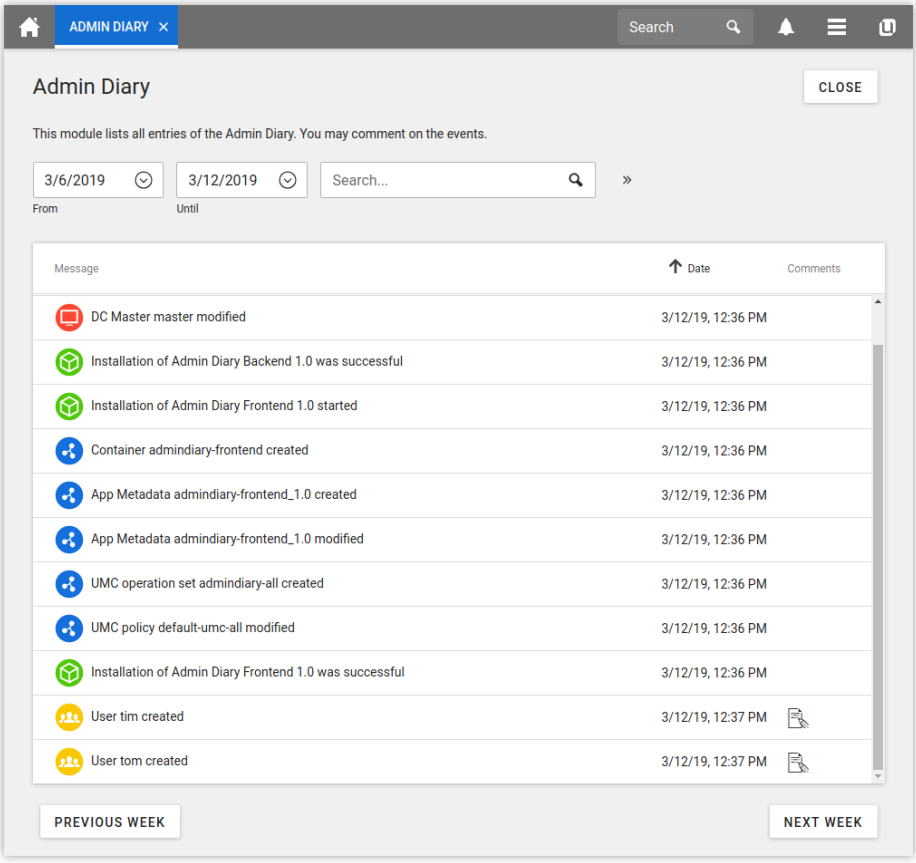
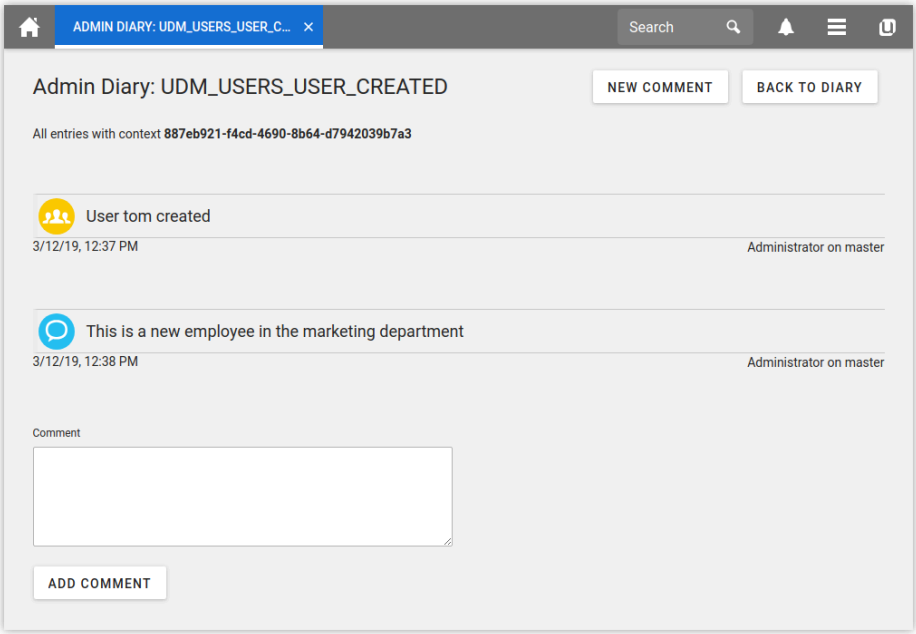



Figure 3.5 shows, how events are shown in the Univention Management Console. By default the displayed entries are grouped by week and can additionally be filtered through the search field. Selecting an entry from the list opens a dialog showing additional details about the who and when of the event, as shown in Figure 3.6. Moreover there is the possibility to comment each event.

**Figure 3.6. Detail view in Admin Diary**




The app consists of two components:

Admin Diary backend

The backend must be installed on one system in the domain before the frontend can be installed. It includes a customization for ***rsyslog*** and writes into a central database, which defaults to PostgreSQL. If MariaDB or MySQL is already installed on the target system, it will be used instead of PostgreSQL.

Admin Diary frontend

Likewise the frontend must be installed at least once, but more installations are also possible. The frontend includes the Univention Management Console module, which is used to show and comment the entries. When installing it on a different host than where the backend is installed, access to the central database needs to be configured manually. The required steps for this are described in this separate document[2].

[2] https://help.univention.com/t/admin-diary-how-to-seperate-frontend-and-backend/11331

univention
be open.

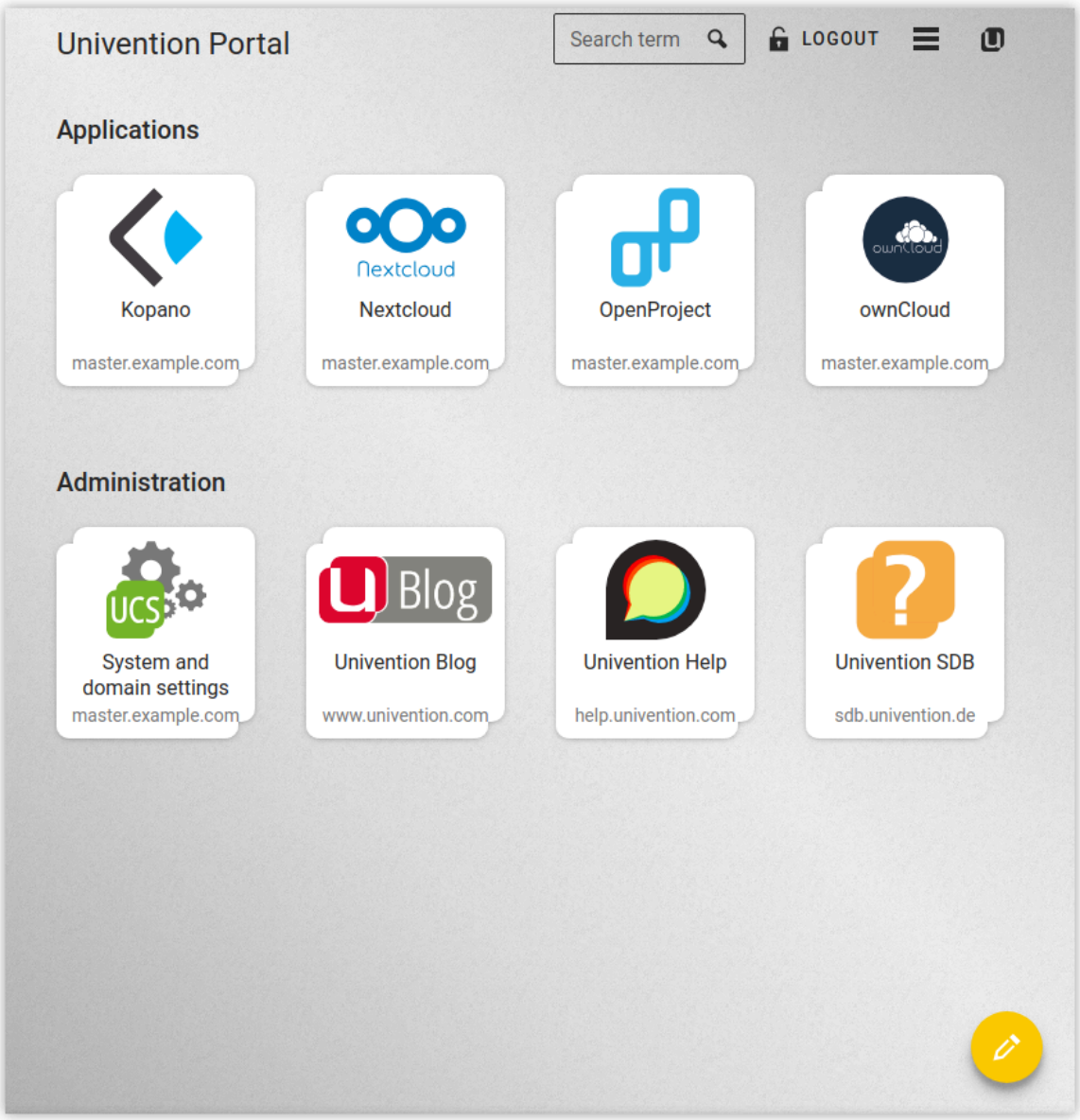
# Chapter 4. UCS web interface


4.1. Introduction ..... 64
4.1.1. Access ..... 65
4.1.2. Browser compatibility ..... 65
4.1.3. Feedback on UMC and UCS ..... 65
4.1.4. Collection of usage statistics ..... 66
4.2. Login ..... 66
4.3. UCS portal page ..... 67
4.3.1. Assign rights for portal settings ..... 68
4.4. Univention Management Console ..... 69
4.4.1. Introduction ..... 69
4.4.2. Activation of UCS license / license overview ..... 69
4.4.3. Operating instructions for modules to administrate LDAP directory data ..... 70
4.4.3.1. Searching for objects ..... 72
4.4.3.2. Creating objects ..... 73
4.4.3.3. Editing objects ..... 73
4.4.3.4. Deleting objects ..... 73
4.4.3.5. Moving objects ..... 73
4.4.4. Favorites ..... 73
4.4.5. Display of system notifications ..... 73
4.5. LDAP directory browser ..... 73
4.6. Policies ..... 75
4.6.1. Creating a policy ..... 75
4.6.2. Applying policies ..... 75
4.6.3. Editing a policy ..... 76
4.7. Expansion of UMC with extended attributes ..... 76
4.8. Structuring of the domain with user-defined LDAP structures ..... 80
4.9. Delegated administration in the UMC ..... 81
4.10. Command line interface of domain management (Univention Directory Manager) ..... 81
4.10.1. Parameters of the command line interface ..... 82
4.10.2. Example invocations of the command line interface ..... 84
4.10.2.1. Users ..... 84
4.10.2.2. Groups ..... 85
4.10.2.3. Container / Policies ..... 85
4.10.2.4. Computers ..... 86
4.10.2.5. Shares ..... 86
4.10.2.6. Printers ..... 86
4.10.2.7. DNS/DHCP ..... 87
4.10.2.8. Extended attributes ..... 87
4.11. HTTP API of domain management ..... 88
4.12. Evaluation of data from the LDAP directory with Univention Directory Reports ..... 88
4.12.1. Creating reports in Univention Management Console ..... 88
4.12.2. Creating reports on the command line ..... 89
4.12.3. Adjustment/expansion of Univention Directory Reports ..... 89

# 4.1. Introduction

Feedback

**Figure 4.1. UCS portal page**




The UCS web interface is the central tool for managing a UCS domain as well as for accessing installed applications of the domain.

The UCS web interface is divided into several subpages which all have a similarly designed header. Via the symbols in the top right, one may launch a search on the current page (magnifier), log in/out (lock) or open the user menu (three bars). The login at the web interface is done via a central page once for all sub pages of UCS as well as for third party applications as far as a web based *single sign-on* is supported (Section 4.2).

Central starting point for users and administrators for all following actions is the UCS portal page (cf. Figure 4.1). By default, the portal page is available on the master domain controller. It allows an overview of all Apps and further services which are installed in the UCS domain. All other system roles show by default links

to locally installed Apps as well as a link back to the portal page of the master domain controller. All aspects of the portal page can be customized to match one's needs (Section 4.3).

For environments with more than one server, an additional entry to a server overview page is shown on the portal page. This sub page gives an overview of all available UCS systems in the domain. It allows a fast navigation to other systems in order to adjust local settings via Univention Management Console.

Univention Management Console (UMC) is the central tool for web-based administration of the UCS domain. Section 4.4 describes its general operation. There are various modules available for the administration of the different aspects of a domain depending on the respective system role. New UMC modules may be added to a system when installing further software components.

The subsequent sections detail the usage of various aspects of the domain management. Section 4.5 gives an overview of the LDAP directory browser. The use of administrative settings via policies is discussed in Section 4.6. How to extend the scope of function of the domain administration is detailed in Section 4.7. Section 4.8 details how containers and organizational units can be used to structure the LDAP directory. Section 4.9 explains delegating administration rights to additional user groups.

In conclusion, the command line interface of the domain administration is illustrated (Section 4.10), and the evaluation of domain data via the UCS reporting function are explained (Section 4.12).

### 4.1.1. Access

Feedback

The UCS web interface can be opened on any UCS system via the URL `https://servername/`. Alternatively, access is also possible via the server's IP address. Under certain circumstances it may be necessary to access the services over an insecure connection (e.g., if no SSL certificates have been created for the system yet). In this case, `http` must be used instead of `https` in the URL. In this case, passwords are sent over the network in plain text!

### 4.1.2. Browser compatibility

Feedback

The UCS web interface uses numerous JavaScript and CSS functions. Cookies need to be permitted in the browser. The following browsers are supported:

- Chrome as of version 71
- Firefox as of version 60
- Microsoft Edge as of version 18
- Safari and Safari Mobile as of version 12

Users with older browsers may experience display problems.

The UCS web interface is available in German and English (and French if it is chosen as language during the installation from DVD); the language to be used can be changed via the entry **Switch language** of the user menu in the upper right corner.

### 4.1.3. Feedback on UMC and UCS

Feedback

By choosing the **Help -> Feedback** option in the upper right user menu, you can provide feedback on UCS via a web form.

### 4.1.4. Collection of usage statistics

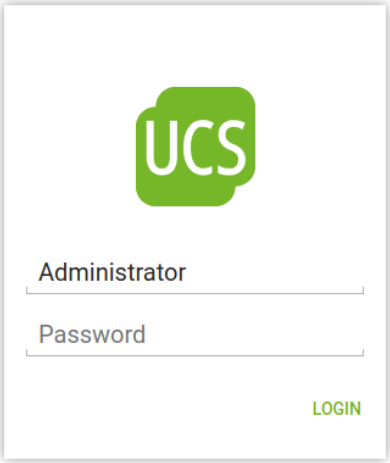
Feedback

Anonymous usage statistics on the use of the UCS web interface are collected when using the *core edition* version of UCS (which is generally used for evaluating UCS). Further information can be found in SDB 1318.

## 4.2. Login

Feedback

**Figure 4.2. UCS login page**




UCS comes with a central login page. By default, a *single sign-on* (SSO) is carried out via SAML (cf. Section 3.9) provided that `ucs-sso.`*`domainname`* can be reached. After successful login, a session is valid for all UCS systems of the domain as well as for third party Apps if these support web based SSO. In case `ucs-sso.`*`domainname`* cannot be reached, the login is carried out at the local UCS system. The session is then only valid for the UCS web pages on the same system. It is possible to enforce a login on the local system by clicking on the link **Login without Single Sign On**.

In the login mask, enter the **Username** and **Password** of the corresponding domain account:

- When logging in with the `Administrator` account on a master domain controller or backup domain controller, UMC modules for the administration and configuration of the local system as well as UMC modules for the administration of data in the LDAP directory are displayed. The initial password of this account has been specified in the setup wizard during the installation. It corresponds to the initial password of the local `root` account. `Administrator` is also the account which should be used for the initial login at a newly installed master domain controller system.

- In some cases, it might be necessary to log on with the system's local `root` account (see Section 8.4.1). This account enables access only to the UMC modules for the administration and configuration of the local system.

- When logging on with another user account, the UMC modules approved for the user are shown. Additional information on allowing further modules can be found in Section 4.9.

The duration of a browser session is 8 hours for the SSO login. After these, the login process must be carried out again. For the login at the local UCS system, the browser session will be automatically closed after an inactivity of 8 hours.

http://sdb.univention.de/1318

By installing a third-party application, such as privacyIDEA, it is possible to extend the Univention Management Console authentication with a two-factor authentication (2FA). These extensions can be installed from the Univention App Center.

## 4.3. UCS portal page

Feedback

Portal pages offer a central view of all available services in a UCS domain. Requirements strongly differ from small to large environments in organizations, public authorities, or even schools. Therefore, UCS implemented a very flexible and individually customizable concept for portal pages.

As illustrated in Figure 4.3, portal entries (i.e., links to applications/Apps/services; UDM object type `settings/portal_entry`) can be assigned to none, one or multiple portals. A portal itself (UDM object type `settings/portal`) renders all entries which are assigned to it. A portal can then be assigned to none, one or multiple computer objects (this is done at the computer object itself).

By default, UCS is configured to have two portals. The portal *domain* is assigned to the master and backup domain controllers of the domain. In addition to all installed applications of the domain, links to Univention Management Console as well as the server overview are shown on this portal page. The portal *local* is assigned to all other server roles in the domain. Herein, links to locally installed Apps, to the local Univention Management Console instance, as well as a link back to the master domain controller are shown.

Custom portals and portal entries can be defined and managed either via the UMC module **Portal settings** or directly on the portal site.

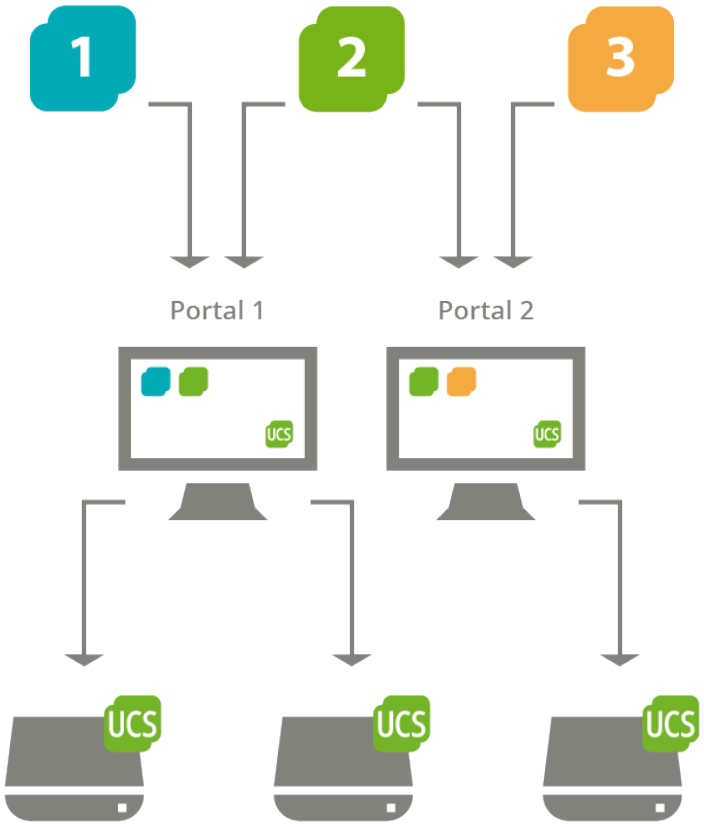
After logging in to the portal on the DC Master or DC Backup, members of the `Domain Admins` group can create new entries on the portal, modify existing entries, modify the order or the design after clicking on the yellow edit icon.

Advanced settings, such as adding new portals or setting which group members can see which portal entries can be made using the UMC portal settings module.

By default, all portal entries are displayed for everyone. In the UMC module **Portal settings** on the tab **Manage anonymous visitors**, it can be configured whether anonymous visitors have to log in before they can see entries. It is also possible to limit certain entries for certain groups. This requires the LDAP attribute *memberOf*. Nested group memberships (i.e., groups in groups) are not evaluated.

Further design adjustments can be made in the file `/usr/share/univention-portal/custom.css`. This file will not be overwritten during an update.

**Figure 4.3. Schema of the portal concept in UCS: Portals can be independently defined and assigned to UCS systems as start site; a link entry can be displayed on multiple portals.**




## 4.3.1. Assign rights for portal settings

Feedback

The following describes how to make the UMC module **Portal settings** accessible to selected groups or users. This example assumes that a group **Portal Admins** has been created and members of this group are supposed to be given access to the portal settings.

On a **domain controller master system** an ACL file has to be created first:

```
/opt/62my-portal-acl.acl
```

This file has to have the following content to allow the necessary ACL changes:

```
access to dn="cn=portal,cn=univention,@%@ldap/base@%@" attrs=children
  by group/univentionGroup/uniqueMember="cn=Portal Admins,cn=groups,@
%@ldap/base@%@" write
  by * +0 break

access to dn.children="cn=portal,cn=univention,@%@ldap/base@%@"
 attrs=entry,@univentionObject,@univentionPortalEntry,
@univentionPortal,@univentionPortalCategory,children
  by group/univentionGroup/uniqueMember="cn=Portal Admins,cn=groups,@
%@ldap/base@%@" write
  by * +0 break
```

Then execute the following command to create an LDAP object for the LDAP ACLs:

```
udm settings/ldapacl create \
 --position "cn=ldapacl,cn=univention,$(ucr get ldap/base)" \
 --set name=62my-portal-acl \
 --set filename=62my-portal-acl \
 --set data="$(bzip2 -c /opt/62my-portal-acl.acl | base64)" \
 --set package="62my-portal-acl" \
 --set packageversion=1
```

If the ACL is to be deleted again, the following command can be used:

```
udm settings/ldapacl remove \
 --dn "cn=62my-portal-acl,cn=ldapacl,cn=univention,$(ucr get ldap/base)"
```

An appropriate UMC policy can now be created via UMC. The following **UMC operations** must be allowed within the policy: **udm-portal**, **udm-syntax**, **udm-validate** and **udm-license**. How to create a policy is described in Section 4.6.1. Now the newly created policy only needs to be assigned to the desired object, in this case the group **Portal Admins**. This can also be done directly within the UMC. For this example, navigate to the group module and edit the desired group there. In the group settings, existing policies for the group object can be selected under **Policies**. More detailed information about policy assignment is described under Section 4.6.2.

# 4.4. Univention Management Console

Feedback

## 4.4.1. Introduction

Feedback

Univention Management Console (UMC) is the central tool for web-based administration of the UCS domain. It can be launched from the portal page (Section 4.3) via the link **System and domain settings** or **System settings**, respectively. Depending on the system role, different UMC modules are provided for administrating UCS. These may be complemented with new modules when installing additional software components.

UMC modules for the administration of all the data included in the LDAP directory (such as users, groups and computer accounts) are only provided on master domain controller and backup domain controller. Changes made in these modules are applied to the whole domain.

UMC modules for the configuration and administration of the local system are provided on all system roles. These modules can be used to install additional applications and updates, adapt the local configuration via Univention Configuration Registry or start/stop services, for example.

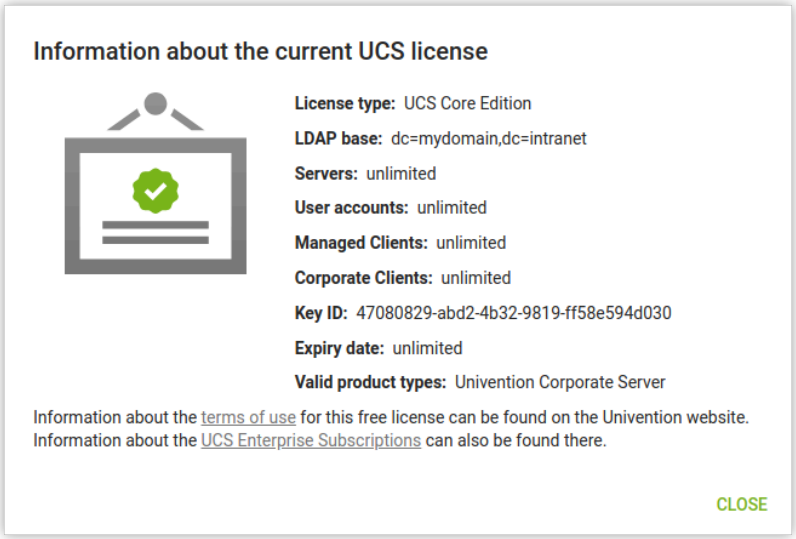
During the first login, an introduction wizard is shown, which informs the user of the collection of usage statistics (see Section 4.1.4) and allows license activation (see Section 4.4.2) among other things.

## 4.4.2. Activation of UCS license / license overview

Feedback

The current license status can be shown on a master domain controller of a domain by clicking on the user menu in the top right line of the screen. Below the menu item **License** the entry **License information** can be selected to open a corresponding information dialogue.

**Figure 4.4. Displaying the UCS license**




The menu entry **Import new license** opens a dialogue in which a new license key can be activated (otherwise the core edition license is used as default license). A license file can be selected and imported via the button **Import from file...**. Alternatively, the license key can also be copied into the input field below and activated with **Import from text field**.

Installation of most of the applications in the Univention App Center requires a personalized license key. UCS core edition licenses can be converted by clicking **Activation of UCS**. The current license key is sent to Univention and the updated key returned to a specified e-mail address within a few minutes. The new key can be imported directly. The conversion does not affect the scope of the license.

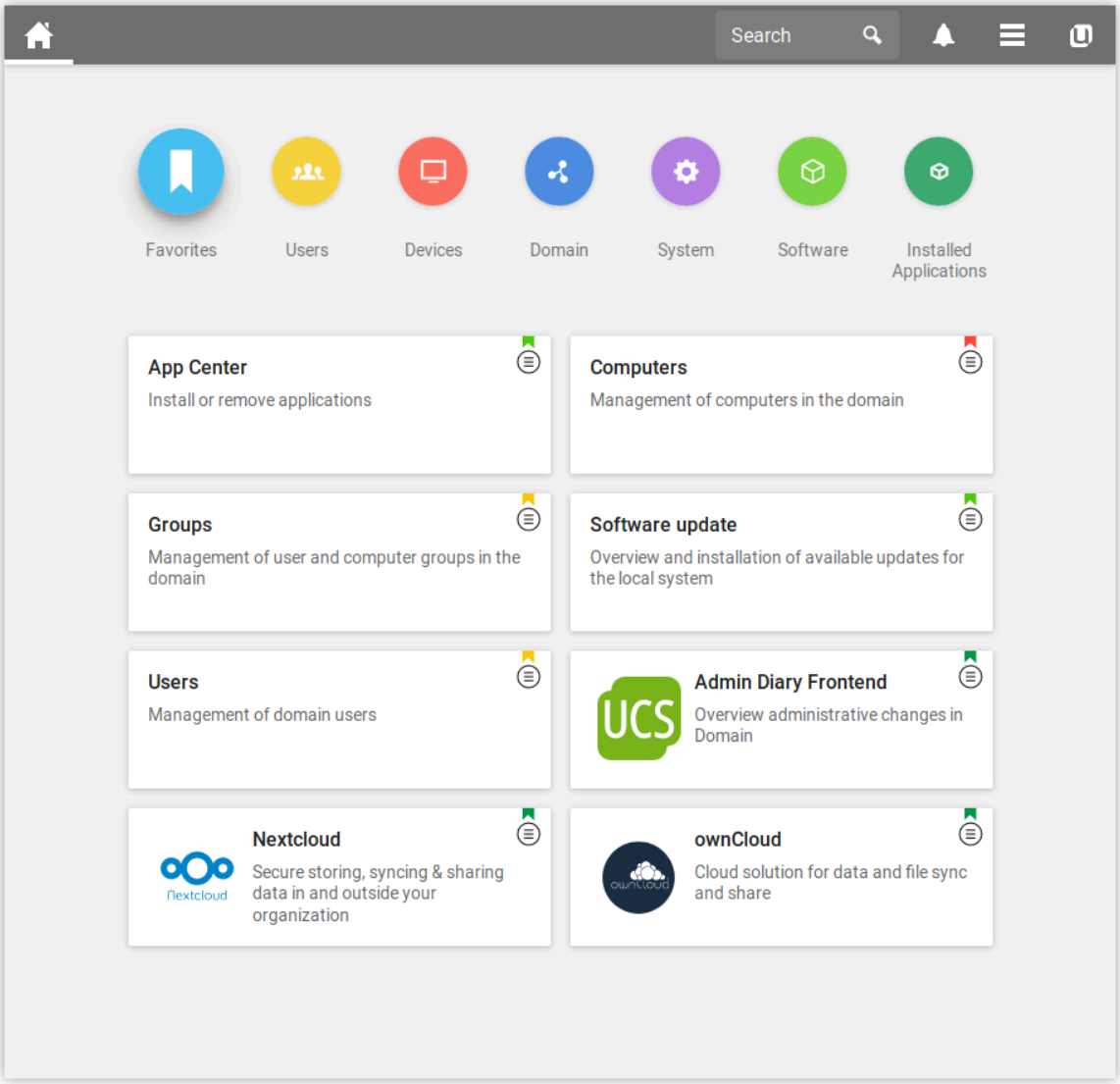
If the number of licensed user or computer objects is exceeded, it is not possible to create any additional objects in Univention Management Console or edit any existing ones unless an extended license is imported or no longer required users or computers are deleted. A corresponding message is displayed on the UMC start page if the license is exceeded.

## 4.4.3. Operating instructions for modules to administrate LDAP directory data

Feedback

All UMC modules for managing LDAP directory objects such as user, group and computer accounts or configurations for printers, shares, mail, Nagios and policies are controlled identically from a structural perspective. The following examples are presented using the user management but apply equally for all modules. The operation of the DNS and DHCP modules is slightly different. Further information can be found in Section 11.2.2 and Section 11.3.2.

**Figure 4.5. Module overview**




The configuration properties/possibilities of the modules are described in the following chapters:

- Users - Chapter 6
- Groups - Chapter 7
- Computers - Chapter 8
- Networks - Section 11.1
- DNS - Section 11.2
- DHCP - Section 11.3
- Shares - Chapter 12
- Printers - Chapter 13
- E-mail - Chapter 14

- Nagios - Section 15.2

The use of policies (Section 4.6) and the LDAP navigation (Section 4.5) are described separately.

### 4.4.3.1. Searching for objects

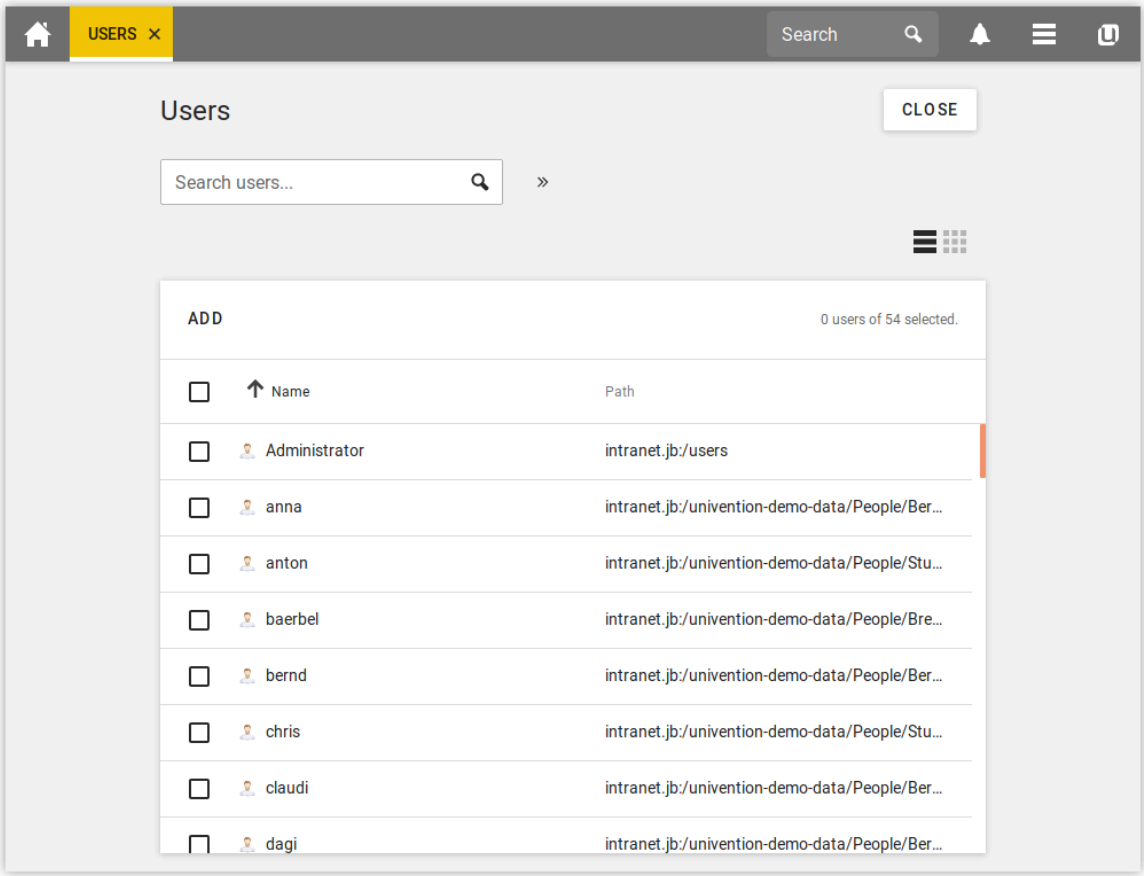
Feedback

The module overview lists all the objects managed by this module. **Search** performs a search for a selection of important attributes (e.g., for user objects by first and last name, primary e-mail address, description, employee number and user name). A wildcard search is also possible, e.g., *m**.

Clicking on **Advanced options** displays additional search options:

- The **Search in** field can be used to select whether the complete LDAP directory or only individual LDAP containers/OUs are searched. Further information on the structure of the LDAP directory service can be found in Section 4.8.
- The **Property** field can be used to search for a certain attribute directly.
- The majority of the modules administrate a range of types of LDAP objects; the computer management for example administrates different objects for the individual system roles. The search can be limited to one type of LDAP object.
- Some of the internally used user groups and groups (e.g., for domain joins) are hidden in the default setting. If the **Include hidden objects** option is enabled, these objects are also shown.

**Figure 4.6. Searching for users**



#### 4.4.3.2. Creating objects

Feedback

The line above the table with the objects includes an actions toolbar which can be used to create a new object using **Add**.

There are simplified wizards for some UMC modules (users, hosts), in which only the most important settings are requested. All attributes can be shown by clicking on **Advanced**.

#### 4.4.3.3. Editing objects

Feedback

Right-clicking on an LDAP object and selecting **Edit** allows to edit the object. The individual attributes are described in the individual documentation chapters. By clicking on the floppy disk symbol in the colored module bar, all changes are written into the LDAP directory. The X symbol cancels the editing and returns to the previous search view.

In front of every item in the result list is a selection field with which the individual objects can be selected. The selection status is also displayed in the lowest screen line, e.g., **2 users of 102 selected** If more than one object is selected, clicking on the stylized pen in the selection status bar activates the multi edit mode. The same attributes are now shown as when editing an individual object, but the changes are only accepted for the objects where the **Overwrite** tick is activated. Only objects of the same type can be edited.

#### 4.4.3.4. Deleting objects

Feedback

Right-clicking on an LDAP object and selecting **Delete** allows to delete the object. The prompt must be confirmed. Some objects use internal references - e.g., a DNS or DHCP object - can be associated with computer objects. These can also be deleted by selecting the **Delete referring objects** option.

Similar to the selection of multiple objects when editing objects, it is also possible to delete multiple objects at once.

#### 4.4.3.5. Moving objects

Feedback

Right-clicking on an LDAP object and selecting **Move to...** allows to to select an LDAP position to which the object should be moved.

Similar to the selection of multiple objects when editing objects, it is also possible to move multiple objects at once.

### 4.4.4. Favorites

Feedback

Commonly used UMC modules are shown in the category **Favorites**. Clicking on a UMC module with the right mouse button opens a context menu. **Add to favorites** and **Remove from favorites** can be used to mark a UMC module as a favorite or remove it again.

### 4.4.5. Display of system notifications

Feedback

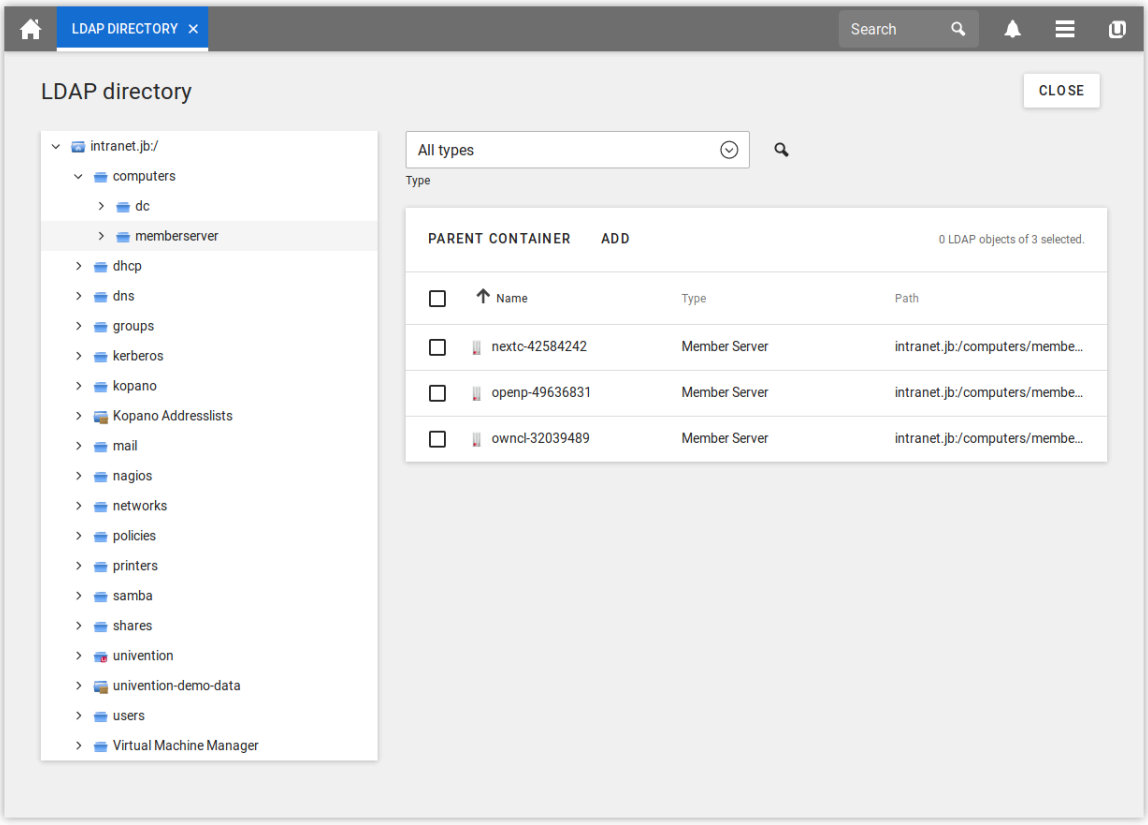
UMC modules can deploy system messages to alert the user to potential errors - e.g., join scripts which have not been run - or necessary actions such as available updates. The messages are shown on the right side and can be dismissed with a mouse click.

## 4.5. LDAP directory browser

Feedback

The **LDAP directory** UMC module can be used to navigate through the LDAP directory. When doing so, new objects can be created, modified or deleted in the LDAP directory.

**Figure 4.7. Navigating the LDAP directory**

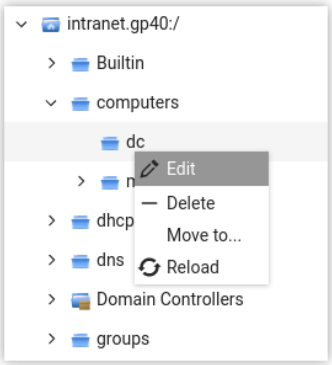



The left half of the screen shows the LDAP directory as a tree structure whose elements can be shown and hidden using the plus and minus buttons.

Clicking on an element of the tree structure switches to this LDAP position and displays the objects at this LDAP position in the overview in the left side of the screen. The selection list **LDAP object type** can be used to limit the display to selected attributes.

The **Add** option can be used to add new objects here too. Similar to the control elements described in Section 4.4, existing objects can also be edited, deleted or moved here.

**Figure 4.8. Editing LDAP container settings**




Right-clicking on an element in the tree structure allows editing the properties of the container or the LDAP base with **Edit**.

# 4.6. Policies

Feedback

*Policies* describe administrative settings which can be practically be used on more than one object. They facilitate the administration as they can be connected to containers and then apply to all the objects in the container in question and the objects in sub containers. The values are applied according to the inheritance principle. For every object, the applied value is always that which lies closest to the object in question.

If, for example, the same password expiry interval is to be defined for all users of a location, then a special container can be created for these users. After moving the user objects into the container, a password policy can be linked to the container. This policy is valid for all user objects within the container.

An exception to this rule is a value which was defined in a policy in the form of **fixed attributes**. Such values cannot be overwritten by subordinate policies.

The command line program `univention-policy-result` can be used to show in detail which policy applies to which directory service object.

Every policy applies to a certain type of UMC domain object, e.g., for users or DHCP subnets.

## 4.6.1. Creating a policy

Feedback

Policies can be managed via the UMC module **Policies**. The operation is the same as for the functions described in Section 4.4.

The attributes and properties of the policies are described in the corresponding chapters, e.g. the DHCP policies in the network chapter.

The names of policies must not contain any umlauts.

**Referencing objects** provides a list of all containers or LDAP objects for which this policy currently applies.

The expanded settings host some general policy options which are generally only required in special cases.

- **Required object classes**: Here you can specify LDAP object classes that an object must possess for the policy to apply to this object. If, for example, a user policy is only relevant for Windows environments, the `sambaSamAccount` object class could be demanded here.
- **Excluded object classes**: Similar to the configuration of the required object classes, you can also list object classes here which should be excluded.
- **Fixed attributes**: Attributes can be selected here, the values of which may not be changed by subordinate policies.
- **Empty attributes**: Attributes can be selected here, which are to be set to empty in the policy, meaning they will be stored without containing a value. This can be useful for removing values inherited by an object from a superordinate policy. In subordinate policies, new values can be assigned to the attributes in question.

## 4.6.2. Applying policies

Feedback

Policies can be assigned in two ways:

- A policy can be assigned to the LDAP base or a container/OU. To do so, the **Policies** tab in the properties of the LDAP object must be opened in the navigation (see Section 4.5).
- A **Policies** tab is shown in the UMC modules of LDAP directory objects for which there are policies available (e.g., for users). A particular policy for a user can be specified at this place.

The **Policies** configuration dialogue is functionally identical; however, all policy types are offered when assigning policies to a LDAP container, whilst only the policy types applicable for the object type in question are offered when assigning policies to an LDAP object.

A policy can be assigned to the LDAP object or container under **Policies**. The values resulting from this policy are displayed directly. The **Inherited** setting means that the settings are adopted from a superordinate policy again - when one exists.

If an object is linked to a policy, or inherits policy settings which cannot be applied to the object, the settings remain without effect for the object. This makes it possible, for example, to assign a policy to the base entry of the LDAP directory, which is then valid for all the objects of the domain which can apply this policy. Objects which cannot apply to this policy are not affected.

### 4.6.3. Editing a policy

Feedback

Policies can be edited and deleted in the UMC module **Policies**. The interface is described in Section 4.4.

#### Caution

When editing a policy, the settings for all the objects linked to this policy are changed! The values from the changed policy apply to objects already registered in the system and linked to the policy, in the same way as to objects added in the future.

The policy tab of the individual LDAP objects also includes the **edit** option, which can be used to edit the policy currently applicable for this object.

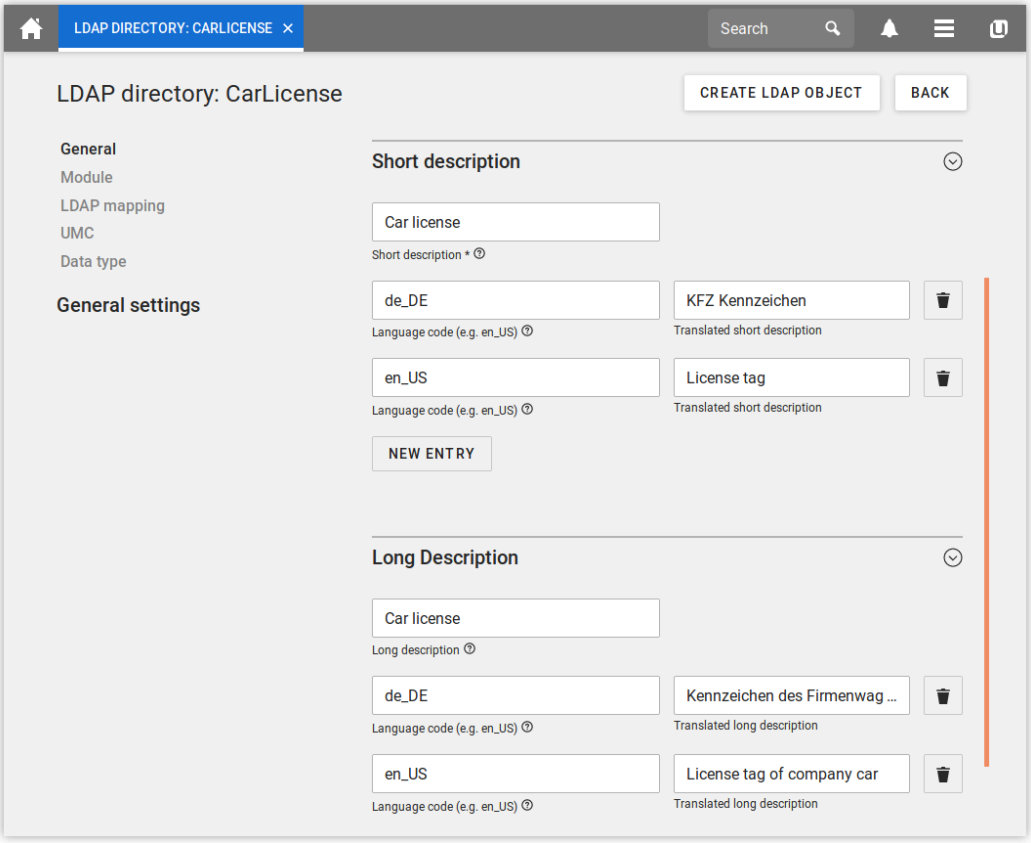
## 4.7. Expansion of UMC with extended attributes

Feedback

The domain management of Univention Management Console allows the comprehensive management of the data in a domain. *Extended attributes* offer the possibility of integrating new attributes in the domain management which are not covered by the UCS standard scope. Extended attributes are also employed by third party vendors for the integration of solutions in UCS.

Extended attributes are managed in the UMC module **LDAP directory**. There one needs to switch to the `univention` container and then to the `custom attributes` subcontainer. Existing attributes can be edited here or a new **Settings: extended attribute** object created here with **Add**.

**Figure 4.9. Extended attribute for managing a car license**




Extended attributes can be internationalized. In this case, the name and description should be compiled in English as this is the standard language for Univention Management Console.

**Table 4.1. 'General' tab**

| Attribute | Description |
|---|---|
| Unique name | The name of the LDAP object which will be used to store the extended attribute. Within a container, the name has to be unique. |
| UDM CLI name | The specified attribute name should be used when employing the command line interface Univention Directory Manager. When the extended attribute is saved, the *Unique name* of the *General* tab is automatically adopted and can be subsequently modified. |
| Short description | Used as title of the input field in Univention Management Console or as the attribute description in the command line interface. |
| Translations of short description | Translated short descriptions can be saved in several languages so that the title of extended attributes is also output with other language settings in the respective national language. This can be done by assigning the respective short description to a language code (e.g., `de_DE` or `fr_FR`) in this input field. |
| Long description | This long description is shown as a tool tip in the input fields in Univention Management Console. |
| Translations of long description | Additional information displayed in the tool tip for an extended attribute can also be saved for several languages. This can be done by assign- |

| Attribute | Description |
|---|---|
| | ing the respective long description to a language code (e.g., `de_DE` or `fr_FR`) in this input field. |

**Table 4.2. 'Module' tab**

| Attribute | Description |
|---|---|
| Modules to be extended | The Univention Directory Manager module which is to be expanded with the extended attribute. An extended attribute can apply for multiple modules. |
| Required options/object classes | Some extended attributes can only be used practically if certain object classes are activated on the *(Options)* tab. One or more options can optionally be saved in this input field so that this extended attribute is displayed or editable. |
| Hook class | The functions of the hook class specified here are used during saving, modifying and deleting the objects with extended attributes. Additional information can be found in [developer-reference]. |

**Table 4.3. 'LDAP' tab**

| Attribute | Description |
|---|---|
| LDAP object class | Object class to which the attribute entered under *LDAP attribute* belongs.<br><br>Predefined LDAP schema extensions for extended attributes are provided with the object class `univentionFreeAttributes`. Further information can be found in Section 3.4.1.1.<br><br>Each LDAP object which should be extended with an attribute is automatically extended with the LDAP object class specified here if a value for the extended attribute has been entered by the user. |
| LDAP attribute | The name of the LDAP attribute where the values of the LDAP object are to be stored. The LDAP attribute must be included in the specified object class. |
| Remove object class if the attribute is removed | If the value of a extended attribute in Univention Management Console is deleted, the attribute is removed from the LDAP object. If no further attributes of the registered object class are used in this LDAP object, the *LDAP object class* will also be removed from the LDAP object if this option is activated. |

**Table 4.4. 'UMC' tab**

| Attribute | Description |
|---|---|
| Do not show this extended attribute in the UMC | This option can be activated if an attribute should only be administrated internally instead of by the administrator, e.g., indirectly by scripts. The attribute can then only be set via the command line interface Univention Directory Manager and is not displayed in the Univention Management Console. |
| Exclude from UMC search | If it should not be possible to search for an extended attribute in the search window of a wizard, this option can be activated to remove the extended attribute from the list of possible search criteria. |

| Attribute | Description |
|---|---|
| | This is only needed in exceptional cases. |
| Ordering number | If several extended attributes are to be managed on one tab, the order of the individual attributes on the tab can be influenced here. They are added to the end of the tab or the group in question in ascending order of their numbers.<br><br>Assigning consecutive position numbers results in the attributes being ordered on the left and right alternately in two columns. Otherwise, the positioning starts in the left column. If additional attributes have the same position number, their order is random. |
| Overwrite existing widget | In some cases it is useful to overwrite predefined input fields with extended attributes. If the internal UDM name of an attribute is configured here, its input field is overwritten by this extended attribute. The UDM attribute name can be identified with the command `univention-directory-manager` (see Section 4.10). This option may cause problems if it is applied to a mandatory attribute. |
| Span both columns | As standard all input fields are grouped into two columns. This option can be used for overlong input fields, which need the full width of the tab. |
| Tab name | The name of the tab in Univention Management Console on which the extended attribute should be displayed. New tabs can also be added here.<br><br>If no tab name is entered, *user-defined* will be used. |
| Translations of tab name | Translated tab names can be assigned to the corresponding language code (e.g. `de_DE` or `fr_FR`) in this input field. |
| Overwrite existing tab | If this option is activated, the tab in question is overwritten before the extended attributes are positioned on it. This option can be used to hide existing input fields on a predefined tab. It must be noted that this option can cause problems with compulsory fields. If the tab to be overwritten uses translations, the overwriting tab must also include identical translations. |
| Tab with advanced settings | Settings possibilities which are rarely used can be placed in the extended settings tab. |
| Group name | Groups allow the structuring of a tab. A group is separated by a gray horizontal bar and can be shown and hidden.<br><br>If no group name is specified for an extended attribute, the attribute is placed above the first group entry. |
| Translations of group name | To translate the name of the group, translated group names for the corresponding language code can be saved in this input field (e.g., `de_DE` or `fr_FR`). |
| Group ordering number | If multiple groups are managed in one tab, this position number can be used to specify the order of the groups. They are shown in the ascending order of their position numbers. |

**Table 4.5. 'Data type' tab**

| Attribute | Description |
|---|---|
| Syntax class | When values are entered, Univention Management Console performs a syntax check.<br><br>Apart from standard syntax definitions (`string`) and (`integer`), there are three possibilities for expressing a binary condition. The syntax `TrueFalse` is represented at LDAP level using the strings `true` and `false`, the syntax `TrueFalseUpper` corresponds to the OpenLDAP boolean values `TRUE` and `FALSE` and the syntax `boolean` does not save any value or the string *1*.<br><br>The syntax `string` is the default. An overview of the additionally available syntax definitions and instructions on integrating your own syntaxes can be found in [developer-reference]. |
| Default value | If a preset value is defined here, new objects to be created will be initialized with this value. The value can still be edited manually during creation. Existing objects remain unchanged. |
| Multi value | This option establishes whether a single value or multiple values can be entered in the input mask. The scheme definition of the LDAP attribute specifies whether one or several instances of the attribute may be used in one LDAP object. |
| Value required | If this option is active, a valid value must be entered for the extended attribute in order to create or save the object in question. |
| Editable after creation | This option establishes whether the object saved in the extended attribute can only be modified when saving the object, or whether it can also be modified subsequently. |
| Value is only managed internally | If this option is activated, the attribute cannot be modified manually, neither at creation time, nor later. This is useful for internal state information configured through a hook function or internally inside a module. |

# 4.8. Structuring of the domain with user-defined LDAP structures

Feedback

Containers and organizational units (OU) are used to structure the data in the LDAP directory. There is no technical difference between the two types, just in their application:

- Organizational units usually represent real, existing units such as a department in a company or an institution
- Containers are usually used for fictitious units such as all the computers within a company

Containers and organizational units are managed in the UMC module **LDAP directory** and are created with **Add** and the object types **Container: Container** and **Container: Organisational unit**.

Containers and OUs can in principle be added at any position in the LDAP; however, OUs cannot be created below containers.

**Table 4.6. 'General' tab**

| Attribute | Description |
|---|---|
| Name | A random name for the container / organizational unit. |

| Attribute | Description |
|---|---|
| Description | A random description for the container / organizational unit. |

**Table 4.7. 'Advanced settings' tab**

| Attribute | Description |
|---|---|
| Add to standard *object type* containers | If this option is activated, the container or organizational unit will be regarded as a standard container for a certain object type. If the current container is declared the standard user container, for example, this container will also be displayed in users search and create masks. |

**Table 4.8. 'Policies' tab**

| Attribute | Description |
|---|---|
| | The tab is described in Section 4.6.2. |

# 4.9. Delegated administration in the UMC

Feedback

In the default setting, only the members of the `Domain Admins` group can access all UMC modules. Policies can be used to configure the access to UMC modules for groups or individual users. For example, this can be used to assign a helpdesk team the authority to manage printers without giving them complete access to the administration of the domain.

UMC modules are assigned via a **UMC** policy which can be assigned to user and group objects. The evaluation is performed additively, i.e., general access rights can be assigned via ACLs assigned to groups and these rights can be extended via ACLs bound to user (see Section 4.6).

In addition to the assignment of UMC policies, LDAP access rights need to be taken into account, as well, for modules that manage data in the LDAP directory. All LDAP modifications are applied to the whole UCS domain. Therefore, in the default setting, only members of the `Domain Admins` group and some internally used accounts have full access to the UCS LDAP. If a module is granted via a UMC policy, the LDAP access must also be allowed for the user/group in the LDAP ACLs. Further information on LDAP ACLs can be found in Section Section 3.4.5.

**Table 4.9. Policy 'UMC'**

| Attribute | Description |
|---|---|
| List of allowed UCS operation sets | All the UMC modules defined here are displayed to the user or group to which this ACL is applied. The names of the domain modules begin with 'UDM'. |

### Caution

For access to UMC, only policies are considered that are assigned to groups or directly to user and computer accounts. Nested group memberships (i.e., groups in groups) are not evaluated.

# 4.10. Command line interface of domain management (Univention Directory Manager)

Feedback

The Univention Directory Manager is the command line interface of the domain management function of Univention Management Console. It expands the web-based interface of the Univention Management Con-

sole and functions as a powerful tool for the automation of administrative procedures in scripts and for the integration in other programs.

Univention Directory Manager can be started with the `univention-directory-manager` command (short form `udm`) as the `root` user on the master domain controller.

Univention Management Console and Univention Directory Manager use the same domain management modules, i.e., all functions of the web interface are also available in the command line interface.

## 4.10.1. Parameters of the command line interface

Feedback

A complete list of available modules is displayed if the `udm` is run with the `modules` parameter:

```
# univention-directory-manager modules
Available Modules are:
  computers/managedclient
  computers/computer
  computers/domaincontroller_backup
  computers/domaincontroller_master
  computers/domaincontroller_slave
 [...]
```

There are up to five operations for every module:

- `list` lists all existing objects of this type
- `create` creates a new object
- `modify` for the editing of existing objects
- `remove` deletes an object
- `move` is used to move an object to another position in the LDAP directory

The possible options of a UDM module and the operations which can be used on it can be output by specifying the operation name, e.g.,

```
univention-directory-manager users/user move
[...]
create options:
  --binddn                        bind DN
  --bindpwd                       bind password
[...]
modify options:
  --binddn                        bind DN
  --bindpwd                       bind password
  --dn                            Edit object with DN
[...]
remove options:
  --binddn                        bind DN
  --bindpwd                       bind password
  --dn                            Remove object with DN
  --arg                           Remove object with ARG
[...]
list options:
  --filter                        Lookup filter
```

```
[...]
move options:
  --binddn                          bind DN
  --bindpwd                         bind password
[...]
```

The following command outputs further information, the operations and the options for every module. This also displays all attributes of the module:

```
univention-directory-manager category/modulename
```

With the `create` operation, the attributes marked with `(*)` must be specified when creating a new object.

Some attributes can be assigned more than one value (e.g., mail addresses to user objects). These multi-value fields are marked with `[]` behind the attribute name. Some attributes can only be set if certain options are set for the object. This is performed for the individual attributes by entering the option name:

```
users/user variables:
  General:
    username (*)                              Username
[...]
  Contact:
    e-mail (person,[])                        E-Mail Address
```

Here, `username (*)` signifies that this attribute must always be set when creating user objects. If the `person` option is set for the user account (this is the standard case), one or more e-mail addresses can be added to the contact information.

A range of standard parameters are defined for every module:

- The parameter `--dn` is used to specify the LDAP position of the object during modifications or deletion. The complete DN must be entered, e.g.,

  ```
  univention-directory-manager users/user remove \
      --dn "uid=ldapadmin,cn=users,dc=company,dc=example"
  ```

- The `--position` parameter is used to specify at which LDAP position an object should be created. If no `--position` is entered, the object is created below the LDAP base! In the `move` operation, this parameter specifies to which position an object should be moved, e.g:

  ```
  univention-directory-manager computers/managedclient move \
      --dn "cn=desk01,cn=management,cn=computers,dc=company,dc=com" \
      --position "cn=finance,cn=computers,dc=company,dc=example"
  ```

- The `--set` parameter specifies that the given value should be assigned to the following attribute. The parameter must be used per attribute value pair, e.g:

  ```
  univention-directory-manager users/user create \
      --position "cn=users,dc=compaby,dc=example" \
      --set username="jsmith" \
      --set firstname="John" \
      --set lastname="Smith" \
      --set password="12345678"
  ```

- `--option` defines the LDAP object classes of an object. If, for example, only `pki` is provided as options for a user object, it is not possible to specify a *mailPrimaryAddress* for this user as this attribute is part of the `mail` option:

- `--superordinate` is used to specify dependent, superordinate modules. A DHCP object, for example, requires a DHCP service object under which it can be stored. This is transferred with the `--superordinate` option.

- The `--policy-reference` parameter allows the assignment of policies to objects (and similarly their deletion with `--policy-dereference`). If a policy is linked to an object, the settings from the policy are used for the object, e.g.:

```
univention-directory-manager category/modulenameOperation\
    --policy-reference "cn=sales,cn=pwhistory,"\
"cn=users,cn=policies,dc=company,dc=example"
```

- The `--ignore_exists` parameters skips existing objects. If it is not possible to create an object, as it already exists, the error code 0 (no error) is still returned.

- `--append` and `--remove` are used to add/remove a value from a multi-value field, e.g.:

```
univention-directory-manager groups/group modify \
    --dn "cn=staff,cn=groups,dc=company,dc=example" \
    --append users="uid=smith,cn=users,dc=company,dc=example" \
    --remove users="uid=miller,cn=users,dc=company,dc=example"
```

## 4.10.2. Example invocations of the command line interface

Feedback

The following examples for the command line front end of Univention Directory Manager can be used as templates for your own scripts:

### 4.10.2.1. Users

Feedback

Creating a user in the standard user container:

```
univention-directory-manager users/user create \
    --position "cn=users,dc=example,dc=com" \
    --set username="user01" \
    --set firstname="Random" \
    --set lastname="User" \
    --set organisation="Example company LLC" \
    --set mailPrimaryAddress="mail@example.com" \
    --set password="secretpassword"
```

Subsequent addition of the postal address for an existing user:

```
univention-directory-manager users/user modify \
    --dn "uid=user01,cn=users,dc=example,dc=com" \
    --set street="Exemplary Road 42" \
    --set postcode="28239" \
    --set city="Bremen"
```

This command can be used to display all the users whose user name begins with *user*:

```
univention-directory-manager users/user list \
    --filter uid=user*
```

Searching for objects with the `--filter` can also be limited to a position in the LDAP directory; in this case, to all users in the container `cn=bremen,cn=users,dc=example,dc=com`:

```
univention-directory-manager users/user list \
    --filter uid="user*" \
    --position "cn=bremen,cn=users,dc=example,dc=com"
```

This call removes the user `user04`:

```
univention-directory-manager users/user remove \
    --dn "uid=user04,cn=users,dc=example,dc=com"
```

A company has two sites with containers created for each. The following command can be used to transfer a user from the container for the site "Hamburg" to the container for the site "Bremen":

```
univention-directory-manager users/user move \
    --dn "uid=user03,cn=hamburg,cn=users,dc=example,dc=com" \
    --position "cn=bremen,cn=users,dc=example,dc=com"
```

#### 4.10.2.2. Groups

Feedback

Creating a group `Example Users` and adding the user `user01` to this group:

```
univention-directory-manager groups/group create \
    --position "cn=groups,dc=example,dc=com" \
    --set name="Example Users" \
    --set users="uid=user01,cn=users,dc=example,dc=com"
```

Subsequent addition of the user `user02` to the existing group:

```
univention-directory-manager groups/group modify \
    --dn "cn=Example Users,cn=groups,dc=example,dc=com" \
    --append users="uid=user02,cn=users,dc=example,dc=com"
```

**Caution**

A `--set` on the attribute *users* overwrites the list of group members in contrast to `--append`.

Subsequent removal of the user `user01` from the group:

```
univention-directory-manager groups/group modify \
    --dn "cn=Example Users,cn=groups,dc=example,dc=com" \
    --remove users="uid=user01,cn=users,dc=example,dc=com"
```

#### 4.10.2.3. Container / Policies

Feedback

This call creates a container `cn=Bremen` beneath the standard container `cn=computers` for the computers at the "Bremen" site. The additional option *computerPath* also registers this container directly as the standard container for computer objects (see Section 4.8):

```
univention-directory-manager container/cn create \
    --position "cn=computers,dc=example,dc=com" \
    --set name="bremen" \
    --set computerPath=1
```

This command creates a disk quota policy with soft and hard limits and the name *Default quota*:

```
univention-directory-manager policies/share_userquota create \
```

```
    --position "cn=policies,dc=example,dc=com" \
    --set name="Default quota" \
    --set softLimitSpace=5GB \
    --set hardLimitSpace=10GB
```

This policy is now linked to the user container `cn=users`:

```
univention-directory-manager container/cn modify \
    --dn "cn=users,dc=example,dc=com" \
    --policy-reference "cn=Default quota,cn=policies,dc=example,dc=com"
```

Creating a Univention Configuration Registry policy with which the storage time for log files can be set to one year. One space is used to separate the name and value of the variable:

```
univention-directory-manager policies/registry create \
    --position "cn=config-registry,cn=policies,dc=example,dc=com" \
    --set name="default UCR settings" \
    --set registry="logrotate/rotate/count 52"
```

This command can be used to attach an additional value to the created policy:

```
univention-directory-manager policies/registry modify \
    --dn "cn=default UCR settings,cn=config-registry,"\
"cn=policies,dc=example,dc=com" \
    --append registry='"logrotate/compress" "no"'
```

#### 4.10.2.4. Computers

Feedback

In the following example, a Windows client is created. If this client joins the Samba domain at a later point in time (see Section 3.2.2), this computer account is then automatically used:

```
univention-directory-manager computers/windows create \
    --position "cn=computers,dc=example,dc=com" \
    --set name=WinClient01 \
    --set mac=aa:bb:cc:aa:bb:cc \
    --set ip=192.0.2.10
```

#### 4.10.2.5. Shares

Feedback

The following command creates a share *Documentation* on the server `fileserver.example.com`. As long as `/var/shares/documentation/` does not yet exist on the server, it is also created automatically:

```
univention-directory-manager shares/share create \
    --position "cn=shares,dc=example,dc=com" \
    --set name="Documentation" \
    --set host="fileserver.example.com" \
    --set path="/var/shares/documentation"
```

#### 4.10.2.6. Printers

Feedback

Creating a printer share `LaserPrinter01` on the print server `printserver.example.com`. The properties of the printer are specified in the PPD file, the name of which is given relative to the directory `/usr/share/ppd/`. The connected printer is network-compatible and is connected via the IPP protocol.

```
univention-directory-manager shares/printer create \
```

```
    --position "cn=printers,dc=example,dc=com" \
    --set name="LaserPrinter01"  \
    --set spoolHost="printserver.example.com" \
    --set uri="ipp:// 192.0.2.100" \
    --set model="foomatic-rip/HP-Color_LaserJet_9500-Postscript.ppd" \
    --set location="Head office" \
    --set producer="producer: "\
"cn=HP,cn=cups,cn=univention,dc=example,dc=com"
```

**Note**

There must be a blank space between the print protocol and the URL target path in the parameter *uri*. A list of the print protocols can be found in Section 13.4

Printers can be grouped in a printer group for simpler administration. Further information on printer groups can be found in Section 13.5.

```
univention-directory-manager shares/printergroup create \
    --set name=LaserPrinters \
    --set spoolHost="printserver.example.com" \
    --append groupMember=LaserPrinter01 \
    --append groupMember=LaserPrinter02
```

#### 4.10.2.7. DNS/DHCP

Feedback

To configure an IP assignment via DHCP, a DHCP computer entry must be registered for the MAC address. Further information on DHCP can be found in Section 11.3.

```
univention-directory-manager dhcp/host create \
    --superordinate "cn=example.com,cn=dhcp,dc=example,dc=com" \
    --set host="Client222" \
    --set fixedaddress="192.0.2.110" \
    --set hwaddress="ethernet 00:11:22:33:44:55"
```

If it should be possible for a computer name to be resolved via DNS, the following commands can be used to configure a forward (host record) and reverse resolution (PTR record).

```
univention-directory-manager dns/host_record create \
    --superordinate "zoneName=example.com,cn=dns,dc=example,dc=com" \
    --set name="Client222" \
    --set a="192.0.2.110"

univention-directory-manager dns/ptr_record create \
    --superordinate "zoneName=0.168.192.in-addr.arpa,cn=dns,"\
"dc=example,dc=com" \
    --set address="110" \
    --set ptr_record="Client222.example.com."
```

Further information on DNS can be found in Section 11.2.

#### 4.10.2.8. Extended attributes

Feedback

Extended attributes can be used to expand the functional scope of Univention Management Console, see Section 4.7. In the following example, a new attribute is added, where the car license number of the company

car can be saved for each user. The values are managed in the object class `univentionFreeAttributes` created specially for this purpose:

```
univention-directory-manager settings/extended_attribute create \
    --position "cn=custom attributes,cn=univention,dc=example,dc=com" \
    --set name="CarLicense" \
    --set module="users/user" \
    --set ldapMapping="univentionFreeAttribute1" \
    --set objectClass="univentionFreeAttributes" \
    --set longDescription="License plate number of the company car" \
    --set tabName="Company car" \
    --set multivalue=0 \
    --set syntax="string" \
    --set shortDescription="Car license"
```

## 4.11. HTTP API of domain management

Feedback

UCS provides an HTTP API for UDM which can be used to inspect, modify, create and delete UDM objects via HTTP requests.

For more information on the API please refer to [developer-reference].

## 4.12. Evaluation of data from the LDAP directory with Univention Directory Reports

Feedback

Univention Directory Reports offers the possibility of creating predefined reports for any objects to be managed in the directory service.

The structure of the reports is defined using templates. The specification language developed for this purpose allows the use of wildcards, which can be replaced with values from the LDAP directory. Any number of report templates can be created. This allows users to select very detailed reports or just create simple address lists, for example.

The creation of the reports is directly integrated in the web interface of Univention Management Console. Alternatively, the command line program `univention-directory-reports` can be used.

Six report templates are already provided with the delivered Univention Directory Reports, which can be used for users, groups and computers. Three templates create PDF documents and three CSV files, which can be used as an import source for other programs. Further templates can be created and registered.

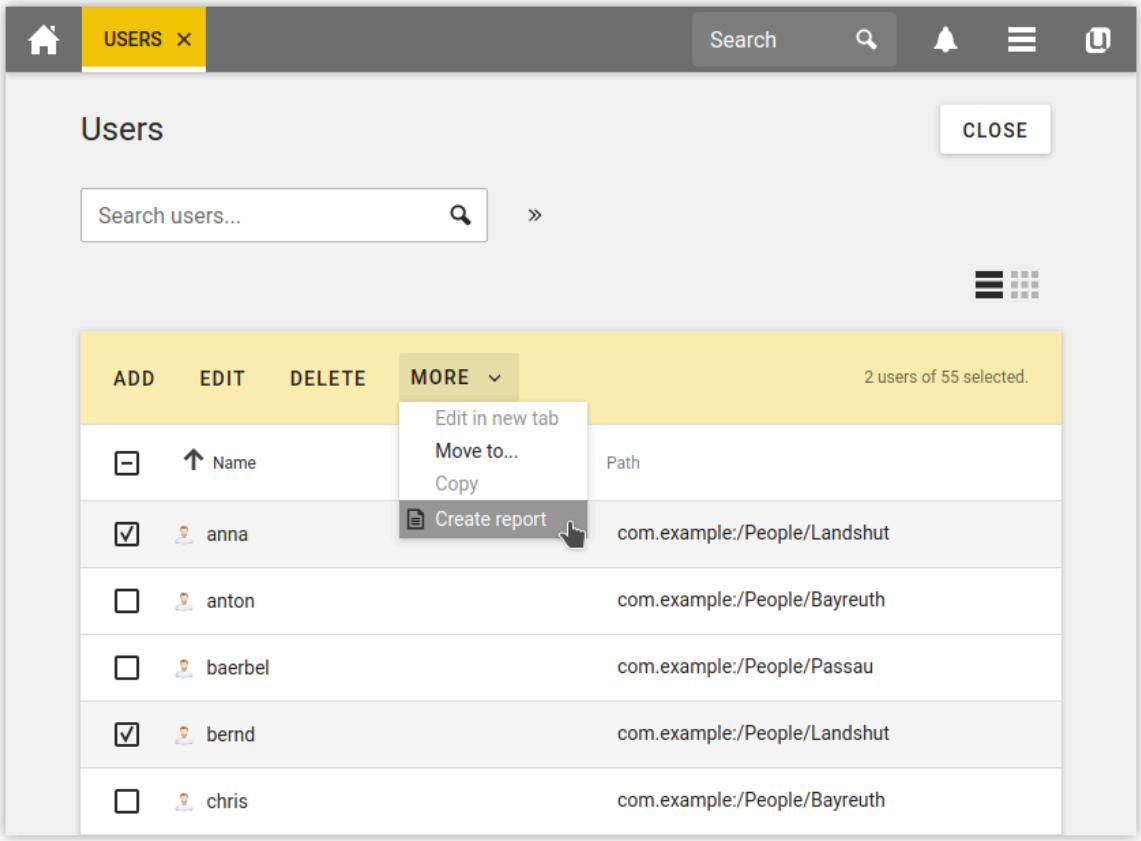
Reports can be created via a command line program or via Univention Management Console.

### 4.12.1. Creating reports in Univention Management Console

Feedback

To create a report, you need to switch to the UMC module for users, groups or hosts. Then all the objects covered by the report must be selected (you can select all objects by clicking the button on the left of **Name**). Clicking on **more -> Create report** allows to choose between the **Standard Report** in PDF format and the **Standard CSV Report** in CSV format.

**Figure 4.10. Creating a report**




The reports created via Univention Management Console are stored for 12 hours and then deleted by a cron job. The settings for when the cron job should run and how long the reports should be stored for can be defined via two Univention Configuration Registry variables:

- `directory/reports/cleanup/cron` Defines when the cron job should be run.
- `directory/reports/cleanup/age`Defines the maximum age of a report document in seconds before it is deleted.

## 4.12.2. Creating reports on the command line

Feedback

Reports can also be created via the command line with the `univention-directory-reports` program. Information on the use of the program can be viewed using the `--help` option.

The following command can be used to list the report templates available to users, for example:

```
univention-directory-reports -m users/user -l
```

## 4.12.3. Adjustment/expansion of Univention Directory Reports

Feedback

Existing reports can be created directly with the presettings. Some presettings can be adapted using Univention Configuration Registry. For example, it is possible to replace the logo that appears in the header of each page of a PDF report. To do so, the value of the Univention Configuration Registry variable `directory/reports/logo` can include the name of an image file. The usual image formats such as JPEG, PNG and GIF can be used. The image is automatically adapted to a fixed width of 5.0 cm.

In addition to the logo, the contents of the report can also be adapted by defining new report templates.

# Chapter 5. Software deployment

5.1. Introduction ..... 91
5.2. Differentiation of update variants / UCS versions ..... 91
5.3. Univention App Center ..... 92
5.4. Updates of UCS systems ..... 96
5.4.1. Update strategy in environments with more than one UCS system ..... 96
5.4.2. Updating individual systems via Univention Management Console ..... 97
5.4.3. Updating individual systems via the command line ..... 98
5.4.4. Updating systems via a policy ..... 98
5.4.5. Postprocessing of release updates ..... 99
5.4.6. Troubleshooting in case of update problems ..... 99
5.5. Configuration of the repository server for updates and package installations ..... 99
5.5.1. Configuration via Univention Management Console ..... 99
5.5.2. Configuration via Univention Configuration Registry ..... 99
5.5.3. Policy-based configuration of the repository server ..... 100
5.5.4. Creating and updating a local repository ..... 100
5.6. Installation of further software ..... 100
5.6.1. Installation/uninstallation of UCS components in the Univention App Center ..... 101
5.6.2. Installation/removal of individual packages in Univention Management Console ..... 101
5.6.3. Installation/removal of individual packages in the command line ..... 102
5.6.4. Hook scripts for administrators ..... 103
5.6.5. Policy-based installation/uninstallation of individual packages via package lists ..... 103
5.7. Specification of an update point using the package maintenance policy ..... 104
5.8. Central monitoring of software installation statuses with the software monitor ..... 104

## 5.1. Introduction

Feedback

The software deployment integrated in UCS offers extensive possibilities for the rollout and updating of UCS installations. Security and version updates can be installed via Univention Management Console, a command line tool and policy-based. This is described in the Section 5.4. The UCS software deployment does not support the updating of Microsoft Windows systems. An additional Windows software distribution is required for this.

For larger installations, there is the possibility of establishing a local repository server from which all further updates can be performed (see Section 5.5). This repository server either procures its packages from the Univention online repository or, in environments without Internet access, also from offline updates in the form of ISO images.

The UCS software deployment is based on the underlying Debian package management tools, which are expanded through UCS-specific tools. The different tools for the installation of software are introduced in Section 5.6. The installation of version and errata updates can be automated via policies, see Section 5.7

The software monitor provides a tool with which all package installations statuses can be centrally stored in a database, see Section 5.8.

The initial installation of UCS systems is not covered in this chapter, but is documented in Chapter 2 instead.

## 5.2. Differentiation of update variants / UCS versions

Feedback

Four types of UCS updates are differentiated:

- *Major releases* appear approximately every three to four years. Major releases can differ significantly from previous major releases in terms of their scope of services, functioning and the software they contain.

- During the maintenance period of a major release, *minor Releases* are released approx. every 10-12 months. These updates include corrections to recently identified errors and the expansion of the product with additional features. At the same time and as far as this is possible, the minor releases are compatible with the previous versions in terms of their functioning, interfaces and operation. Should a change in behavior prove practical or unavoidable, this will be noted in the release notes when the new version is published.

- Univention continuously releases *errata updates*. Errata updates provide fixes for security vulnerabilities and bugfixes/smaller enhancements to make them available to customer systems quickly. An overview of all errata updates can be found at https://errata.software-univention.de/.

- *Patchlevel releases* are released approx. every three months and combine all errata updates published until then.

Every released UCS version has an unambiguous version number; it is composed of a figure (the major version), a full stop, a second figure (the minor version), a hyphen and a third figure (the patch level version). The version UCS 4.2-1 thus refers to the first patch level update for the second minor update for the major release UCS 4.

The *pre-update script* `preup.sh` is run before every release update. It checks for example whether any problems exist, in which case the update is canceled in a controlled manner. The *post-update script* `postup.sh` is run at the end of the update to perform additional cleanups, if necessary.

Errata updates always refer to certain minor releases, e.g., for UCS 4.4. Errata updates can generally be installed for all patch level versions of a minor release.

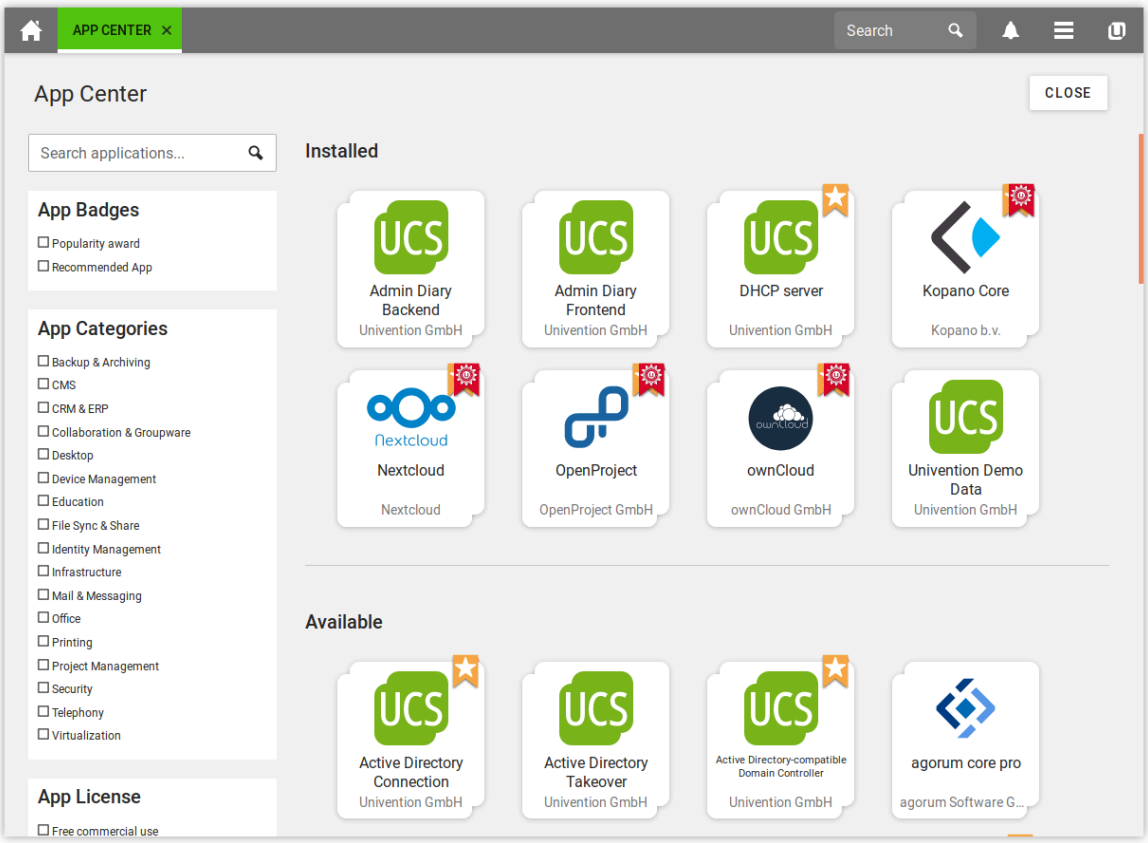
If new release or errata updates are available, a corresponding notification is given when a user logs on to Univention Management Console. The availability of new updates is also notified via e-mail; the corresponding newsletters - separated into release and error updates - can be subscribed on the Univention website. A changelog document is published for every release update listing the updated packages, information on error corrections and new functions and references to the Univention Bugzilla.

## 5.3. Univention App Center

Feedback

The Univention App Center allows simple integration of software components in a UCS domain. The applications are provided both by third parties and by Univention itself (e.g., UCS@school). The maintenance and support for the applications are provided by the respective manufacturer.

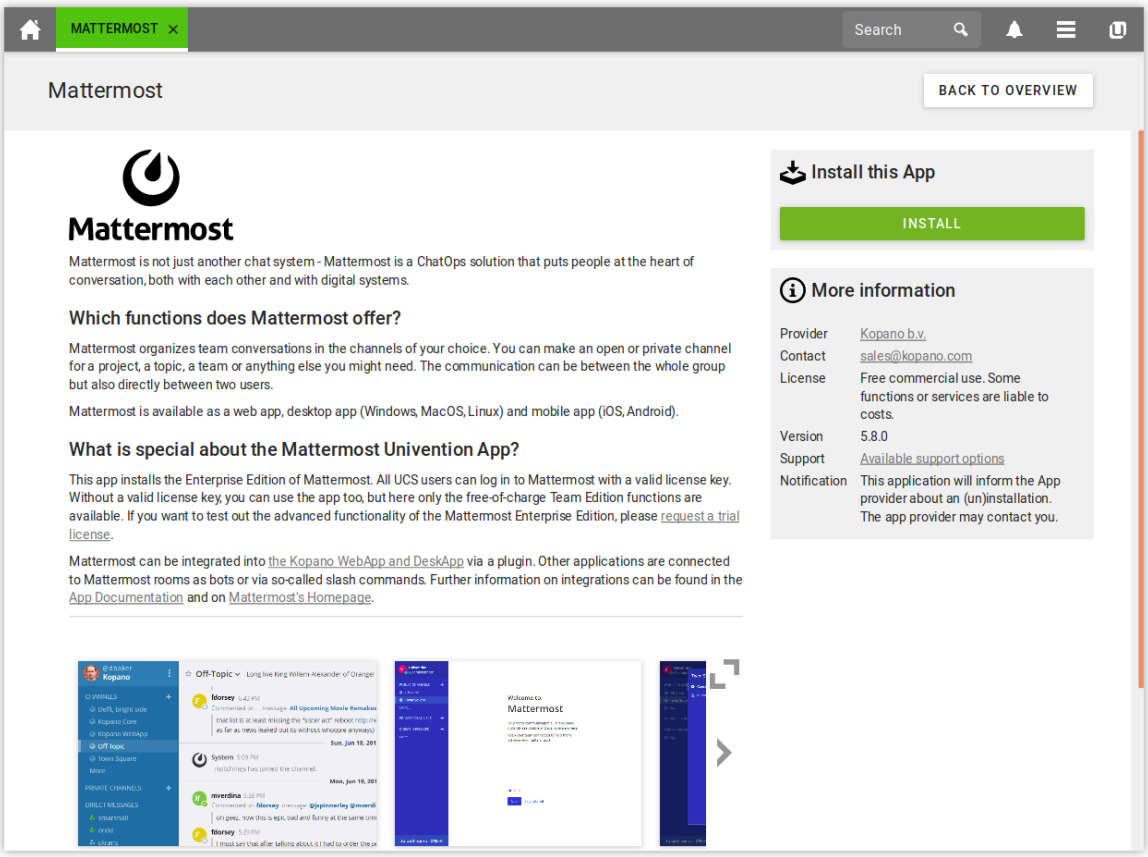
**Figure 5.1. Overview of applications available in the App Center**




The Univention App Center can be opened via the UMC module *App Center*. It shows by default all installed as well as available software components. **Search applications...** can be used to search for available applications. Furthermore, the applications can also be filtered using the **Categories**. More filters like the **App Badges** and the **App License** can be used. For example, the view can be limited to applications with the categories **`Education`** or **`Office`**. To only show the *Recommended Apps* for theses categories, it is sufficient to activate the appropriate filter.

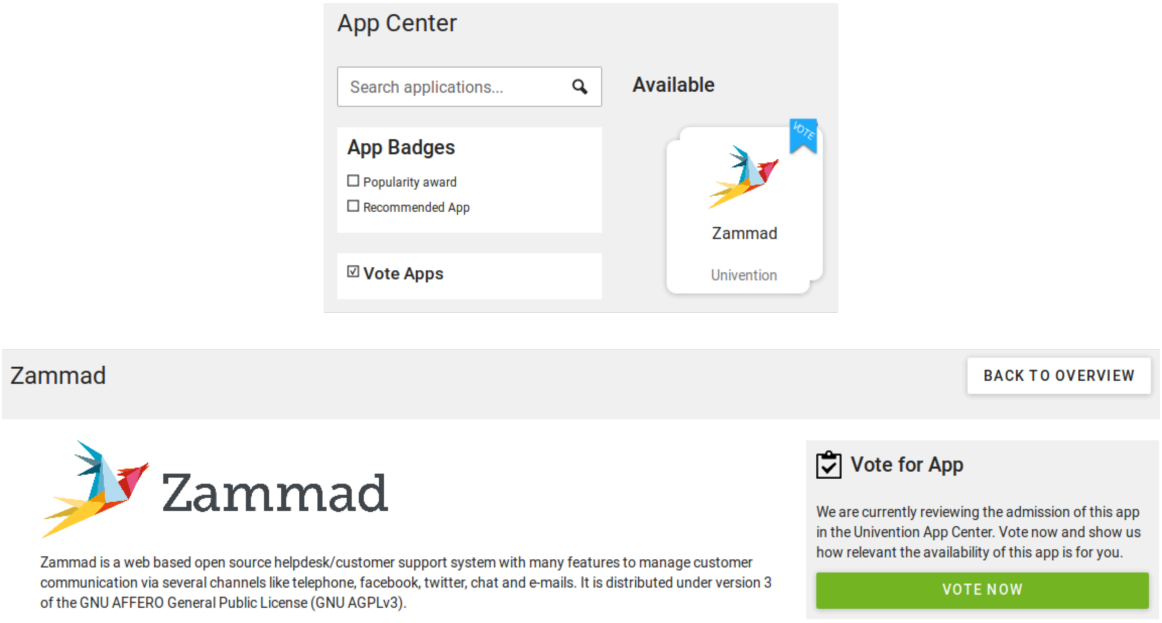
If you click on one of the displayed applications, further details on it are shown (e.g., description, manufacturer, contact information and screenshots or videos). The **Notification** field displays whether the manufacturer of the software component is notified when it is installed/uninstalled. A rough classification of the licensing can be found under the **License** section. Some applications provide a **Buy** button with a link to detailed licensing information. For all other applications, it is recommended to contact the the manufacturer of the application about detailed licensing information using the e-mail address shown under **Contact**.

**Figure 5.2. Details for an application in the App Center**




With *Vote Apps* there is a special form of Apps in the App Center that do not install anything on the UCS system. Voting helps Univention and the potential app provider to determine the interest in this app. Vote apps are usually only displayed for a limited voting period. That Vote Apps are available, can be recognized by the shown **Vote Apps** filter option in the App Center overview.

**Figure 5.3. Example Vote Apps in App Center overview and detail view**




Some applications may not be compatible with other software packages from UCS. For instance, most groupware packages require the UCS mailstack to be uninstalled. Every application checks whether incompatible versions are installed and then prompts which **Conflicts** exist and how they can be resolved. The installation of these packages is then prevented until the conflicts have been resolved.

Some components integrate packages that need to be installed on the master domain controller (usually LDAP schema extensions or new modules for the UCS management system). These packages are automatically installed on the master domain controller. If this is not possible, the installation is aborted. In addition, the packages are set up on all accessible backup domain controller systems. If several UCS systems are available in the domain, it can be selected on which system the application is to be installed.

Some applications use the container technology Docker. In these cases, the application (and its direct environment) is encapsulated from the rest and both security as well as the compatibility with other applications increased.

From a technical perspective, the app is started as Docker container and joined into the UCS domain as member server. A corresponding computer object is created for the member server in the LDAP directory.

On the network side, the container can only be reached from the computer on which the app is installed. The app can, however, open certain ports, which can be forwarded from the actual computer to the container. UCS' firewall is correspondingly configured automatically to allow access to these ports.

If a command line is required in the app's environment, the first step is to switch to the container. This can be done by running the following command (using the fictitious app demo-docker-app as an example in this case):

```
univention-app shell demo-docker-app
```

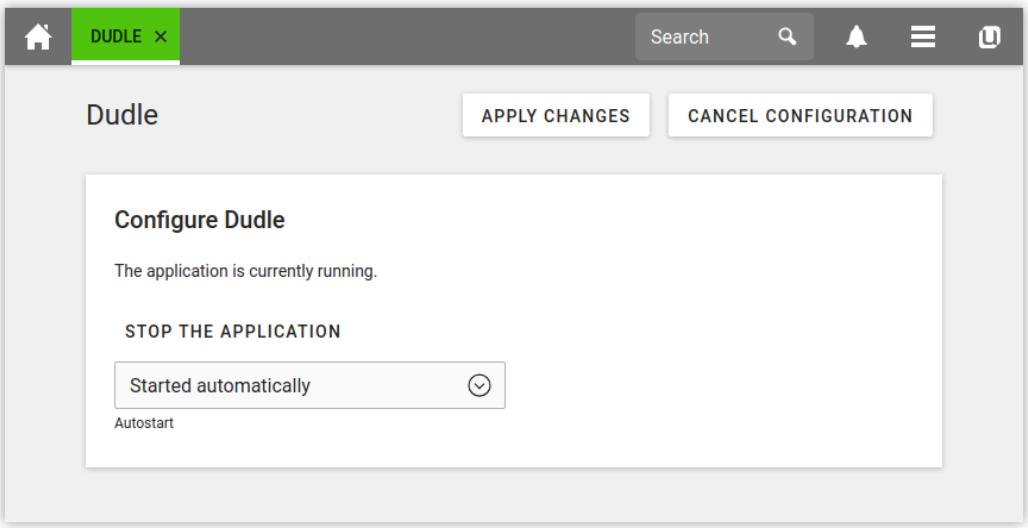
*Docker apps* can be further configured via the UMC module. The app can be started and stopped and the **autostart** option be set:

- Started automatically ensures that the app is started automatically when the server is started up.
- Started manually prevents the app from starting automatically, but it can be started via the UMC module.

- Starting is prevented prevents the app from starting at any time; it cannot even be started via the UMC module.

In addition, apps can also be adjusted using additional parameters. The menu for doing so can be opened using the App Settings button of an installed app.

**Figure 5.4. Setting of an application in the App Center**




After its installation, one or several new options are shown when clicking on the icon of an application: **Uninstall** removes an application. **Open** refers you to a website or a UMC module with which you can further configure or use the installed application. For example, if you install the Horde application, this link takes you to the login window. This option is not displayed for applications which do not have a web interface or a UMC module.

Updates for applications are published independently of the Univention Corporate Server release cycles. If a new version of an application is available, the **Upgrade** menu item is shown, which starts the installation of the new version. If updates are available, a corresponding message is also shown in the UMC module **Software update**. An overview of the installed applications in the domain can be opened under **Installed applications** on the UMC start page.

Installations and the removal of packages are documented in the `/var/log/univention/management-console-module-appcenter.log` log file.

# 5.4. Updates of UCS systems

Feedback

There are two ways to update UCS systems; either on individual systems (via Univention Management Console or command line) or via a Univention Management Console computer policy for larger groups of UCS systems.

## 5.4.1. Update strategy in environments with more than one UCS system

Feedback

In environments with more than one UCS system, the update order of the UCS systems must be borne in mind:

The authoritative version of the LDAP directory service is maintained on the master domain controller and replicated on all the remaining LDAP servers of the UCS domain. As changes to the LDAP schemes (see

Section 3.4.1) can occur during release updates, the master domain controller must always be the first system to be updated during a release update.

It is generally advisable to update all UCS systems in one maintenance window whenever possible. If this is not possible, all not-updated UCS systems should only be one release version older compared with the master domain controller.

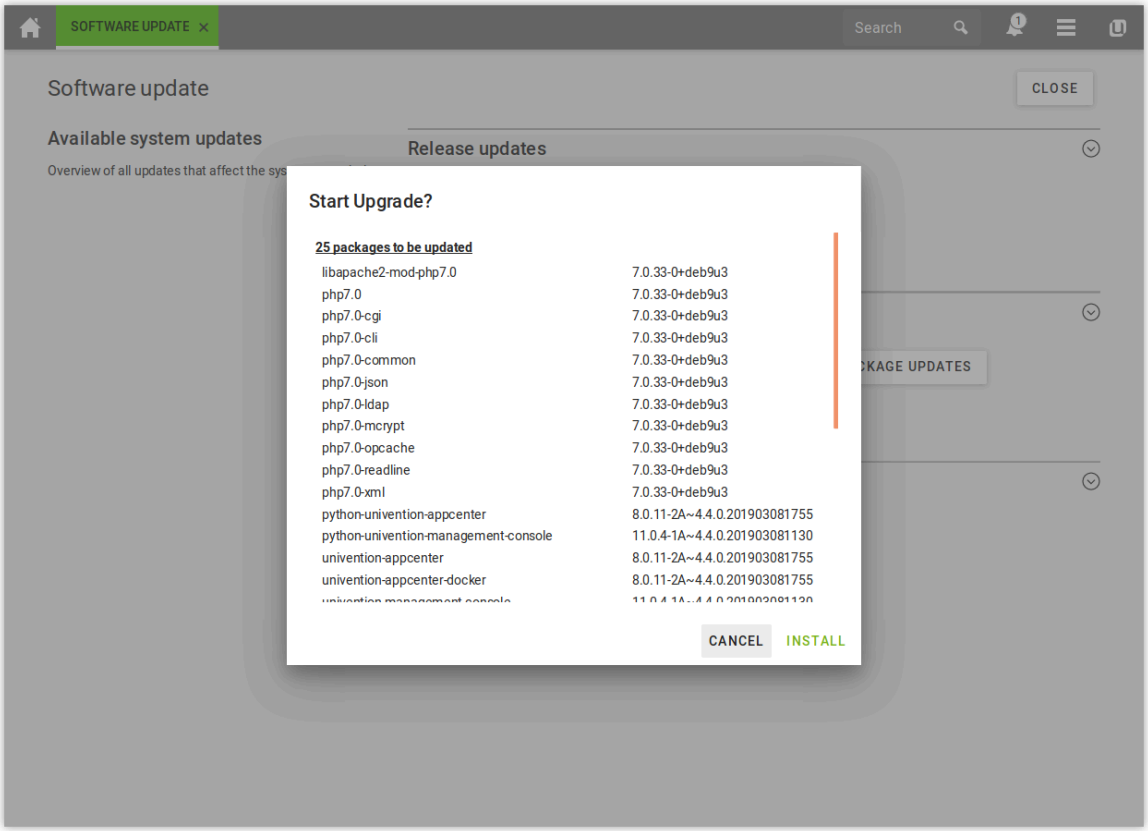
## 5.4.2. Updating individual systems via Univention Management Console

Feedback

The **Software update** module allows the installation of release updates and errata updates.

Figure Figure 5.5 shows the overview page of the module. The currently installed version is displayed under **Release updates** in the upper part of the dialogue box.

**Figure 5.5. Updating a UCS system in UMC**




If a newer UCS version is available, a select list is displayed. After clicking on **Install release updates** and confirmation all updates up to the respective version are installed. Before the installation process is started, a message will be displayed informing the user of possible restrictions of the server services during the update. Any intermediate versions are also installed automatically.

Clicking on **Install available errata updates** installs all the available errata updates for the current release and all installed components.

**Check for package updates** activates an update of the package sources currently entered. This can be used, for example, if an updated version is provided for a component.

The messages created during the update are written to the file `/var/log/univention/updater.log`

## 5.4.3. Updating individual systems via the command line

Feedback

The following steps must be performed with `root` rights.

An individual UCS system can be updated using the `univention-upgrade` command in the command line. A check is performed to establish whether new release or application updates are available and these are then installed if a prompt is confirmed. In addition, package updates are also performed (e.g., in the scope of an errata update).

In the basic setting, the packages to be updated are loaded from a repository via the network. If a local repository is used (see Section 5.5.4), release updates can alternatively also be installed via update DVD images, which are either imported as ISO files or from a drive. This is done by running `univention-upgrade` with the parameters `--iso=`*`ISOIMAGEFILE`* or `--cdrom=`*`DRIVE`*.

Remote updating over SSH is not advisable as this may result in the update procedure being aborted. If updates should occur over a network connection nevertheless, it must be verified that the update continues despite disconnection from the network. This can be done, for example, using the tools `screen` and `at`, which are installed on all system roles.

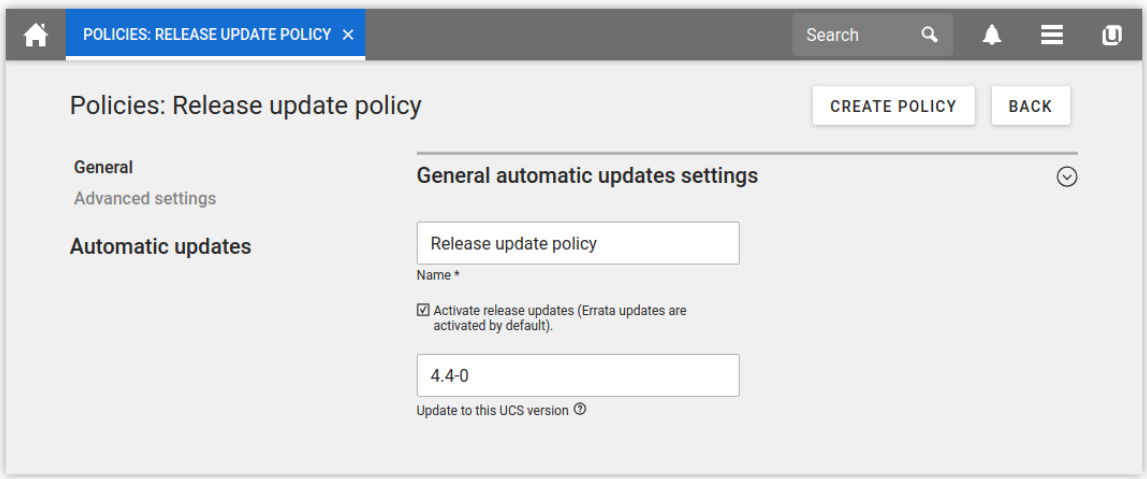
The messages created during the update are written to the file `/var/log/univention/updater.log`

## 5.4.4. Updating systems via a policy

Feedback

An update for more than one computer can be configured with an **Automatic updates** policy in the UMC modules for computer and domain management (see Section 4.6).

**Figure 5.6. Updating UCS systems using an update policy**




A release update is only run when the **Activate release updates** selection field is activated.

The **Update to this UCS version** input field includes the version number up to which the system should be updated, e.g., *4.4-0*. If no entry is made, the system continues updating to the highest available version number.

The point at which the update should be performed is configured via a **Maintenance** policy (see Section 5.7).

The messages created during the update are written to the file `/var/log/univention/updater.log`.

### 5.4.5. Postprocessing of release updates

Feedback

Once a release update has been performed successfully, a check should be made for whether new or updated join scripts need to be run.

Either the *Domain join* UMC module or the command line program `univention-run-join-scripts` is used for checking and starting the join scripts (see Section 3.2.1).

### 5.4.6. Troubleshooting in case of update problems

Feedback

The messages generated during updates are written to the `/var/log/univention/updater.log` file, which can be used for more in-depth error analysis.

The status of the Univention Configuration Registry variables before the release update is saved in the `/var/univention-backup/update-to-TARGETRELEASEVERSION/` directory. This can then be used to check whether and which variables have been changed during the update.

## 5.5. Configuration of the repository server for updates and package installations

Feedback

Package installations and updates can either be performed from the Univention update server or from a locally maintained repository. A local repository is practical if there are a lot of UCS systems to update as the updates only need to be downloaded once in this case. As repositories can also be updated offline, a local repository also allows the updating of UCS environments without Internet access.

Using the registered settings, APT package sources are automatically generated in the `/etc/apt/sources.list.d/` directory for release and errata updates as well as addon components. If further repositories are required on a system, these can be entered in the `/etc/apt/sources.list` file.

In the default setting, the Univention repository `updates.software-univention.de` is used for a new installation.

The Univention repository and repository components differentiate between two component parts:

- The UCS standard package scope covered by maintenance can be found in the *maintained* area. In the default setting, only access to these packages is activated. Security updates are only provided for *maintained* packages.
- Additional packages can be found under *unmaintained*, e.g., other mail servers than Postfix. These packages are not covered by security updates or ulterior maintenance. In the default setting, *unmaintained* is not mounted, but can be integrated by setting the Univention Configuration Registry variable `repository/online/unmaintained` to `yes`.

A local repository can require a lot of disk space - particularly if the unmaintained branch is activated.

### 5.5.1. Configuration via Univention Management Console

Feedback

The **Repository server** and the use of the maintained and unmaintained sections can be specified in the UMC module **Repository Settings**.

### 5.5.2. Configuration via Univention Configuration Registry

Feedback

The repository server to be used can be entered in the Univention Configuration Registry variable `repository/online/server` and is preset to `updates.software-univention.de` for a new installation.

The unmaintained repository can be integrated by setting the Univention Configuration Registry variable `repository/online/unmaintained` to `yes`.

### 5.5.3. Policy-based configuration of the repository server

Feedback

The repository server to be used can also be specified using the **Repository server** policy in the computer management of the Univention Management Console. Only UCS server systems for which a DNS entry has been configured are shown in the selection field (see Section 4.6).

### 5.5.4. Creating and updating a local repository

Feedback

Package installations and updates can either be performed from the Univention update server or from a locally maintained repository. A local repository is practical if there are a lot of UCS systems to update as the updates only need to be downloaded once in this case. As repositories can also be updated offline, a local repository also allows the updating of UCS environments without Internet access.

There is also the possibility of synchronizing local repositories, which means, for example, a main repository is maintained at the company headquarters and then synchronized to local repositories at the individual locations.

To set up a repository, the `univention-repository-create` command must be run as the `root` user. The initial package inventory is imported from an installation DVD. The parameter `--iso` allows importing from an ISO image. UCS is only available as a 64-bit DVD. The repository is created by `univention-repository-create` with the architecture of the specified installation medium. If an environment is operated in which both 32-bit and 64-bit packages are needed, the following commands must be executed on the repository server:

```
ucr set repository/online/architectures="i386 amd64"
univention-repository-update net
```

Access to the Univention online repository is cryptographically secured via the use of Secure APT employing signatures. This feature is not currently available for local repositories and so a message appears when creating a repository explaining how Secure APT can be deactivated using the Univention Configuration Registry variable `update/secure_apt`. This setting must be set on all UCS systems that access the repository.

The packages in the repository can be updated using the `univention-repository-update` tool. It supports two modes:

- `univention-repository-update cdrom`Here the repository is updated with an update DVD or an ISO image.
- `univention-repository-update net` Here the repository is synchronized with another specified repository server. This is defined in the Univention Configuration Registry variable `repository/mirror/server` and typically points to `updates.software-univention.de`.

An overview of the possible options is displayed with the following command:

```
univention-repository-update -h
```

The repository is stored in the `/var/lib/univention-repository/mirror/` directory.

The local repository can be activated/deactivated using the Univention Configuration Registry variable `local/repository`.

## 5.6. Installation of further software

Feedback

The initial selection of the software components of a UCS system is performed within the scope of the installation. The software components are selected relative to the functions, whereby e.g. the *Proxy server* component

is selected, which then procures the actual software packages via a meta package. The administrator does not need to know the actual package names. However, individual packages can also be specifically installed and removed for further tasks. When installing a package, it is sometimes necessary to install additional packages, which are required for the proper functioning of the package. These are called packages dependencies. All software components are loaded from a repository (see Section 5.5).

Software which is not available in the Debian package format should be installed into the `/opt/` or `/usr/local/` directories. These directories are not used for installing UCS packages, thus a clean separation between UCS packages and other software is ensured.

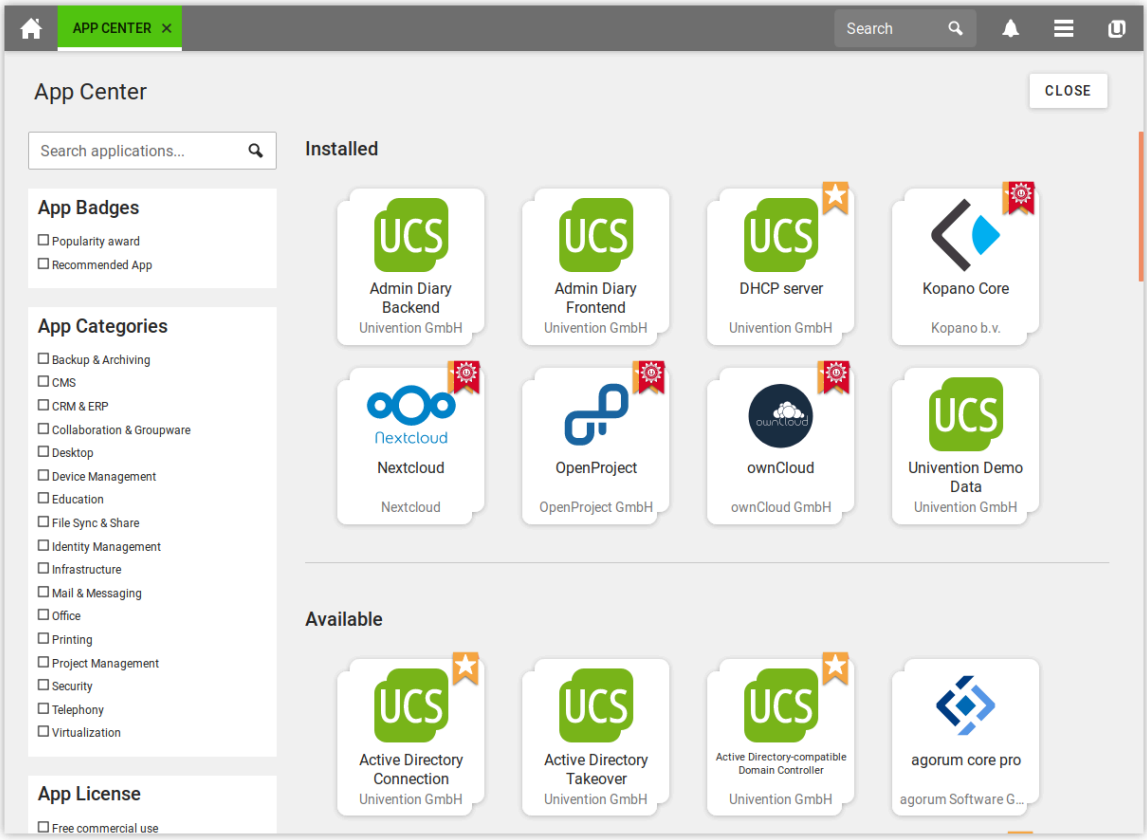
There are several possibilities for installing further packages subsequently on an installed system:

## 5.6.1. Installation/uninstallation of UCS components in the Univention App Center

Feedback

All software components offered in the Univention Installer can also be installed and removed at a later point in time via the Univention App Center. This is done by selecting the **UCS components** package category. Further information on the Univention App Center can be found in Section 5.3.

**Figure 5.7. Selection of UCS components in the App Center**

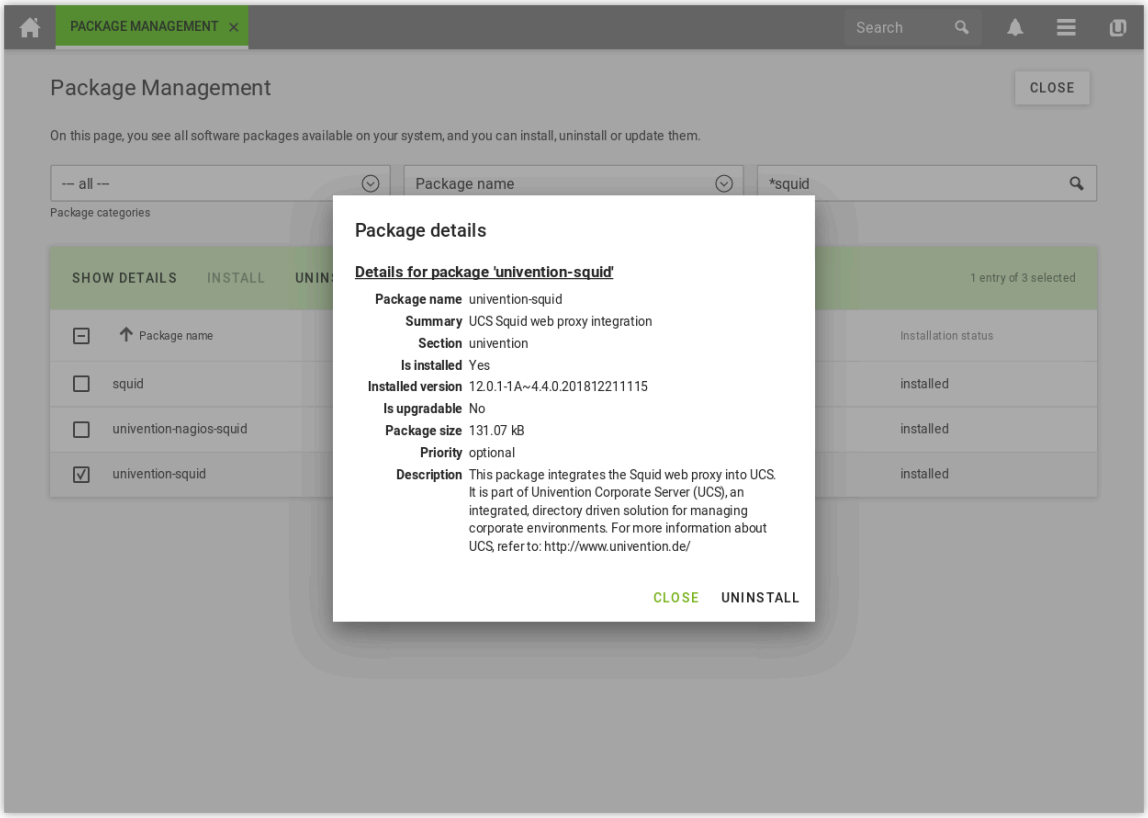



## 5.6.2. Installation/removal of individual packages in Univention Management Console

Feedback

The UMC module **Package Management** can be used to install and uninstall individual software packages.

**Figure 5.8. Installing the package *univention-squid* in Univention Management Console**




A search mask is displayed on the start page in which the user can select the package category or a search filter (name or description). The results are displayed in a table with the following columns:

- Package name
- Package description
- Installation status

Clicking an entry in the result list opens a detailed information page with a comprehensive description of the package.

In addition, one or more buttons will be displayed: **Install** is displayed if the software package is not installed yet; **Uninstall** is displayed if the software package is installed and **Upgrade** is displayed if the software package is installed but not updated. **Close** can be used for returning to the previous search request.

## 5.6.3. Installation/removal of individual packages in the command line

Feedback

The following steps must be performed with `root` rights.

Individual packages are installed using the command

```
univention-install PACKAGENAME
```

Packages can be removed with the following command:

```
univention-remove PACKAGENAME
```

If the name of a package is unknown, the command `apt-cache search` can be used to search for the package. Parts of the name or words which appear in the description of the package are listed, e.g.:

```
apt-cache search fax
```

## 5.6.4. Hook scripts for administrators

Feedback

Custom scripts can be called after each app installation, -upgrade or -removal. Such scripts can be used to automate repeating administrative tasks.

To use this feature custom scripts can be placed in one of the directories listed below. If such a directory does not yet exist, it can be manually created:

```
/var/lib/univention-appcenter/apps/{appid}/local/hooks/post-install.d/
/var/lib/univention-appcenter/apps/{appid}/local/hooks/post-upgrade.d/
/var/lib/univention-appcenter/apps/{appid}/local/hooks/post-remove.d/
```

Where *`{appid}`* is the name of the app for which the scripts should be executed.

Script file names are only allowed to consist of lower case letters and numbers (`^[a-z0-9]+$`). Additionally scripts have to be marked as executable (`chmod +x [filename]`), because they are internally called by `run-parts`. As a consequence `run-parts --test [directory]` can be used to verify if and which files would be executed. Further information can be found in the manual with `man run-parts`.

The `/var/log/univention/appcenter.log` contains possible scripting error messages and further hints.

## 5.6.5. Policy-based installation/uninstallation of individual packages via package lists

Feedback

Package lists can be used to install and remove software using policies. This allows central software deployment for a large number of computer systems.

Each system role has its own package policy type.

Package policies are managed in the UMC module *Policies* with the **Policy: Packages + system role**.

### Table 5.1. 'General' tab

| Attribute | Description |
|---|---|
| Name | An unambiguous name for this package list, e.g., *mail server*. |
| Package installation list | A list of packages to be installed. |
| Package removal list | A list of packages to be removed. |

The software packages defined in a package list are installed/uninstalled at the time defined in the **Maintenance** policy (for the configuration see Section 5.7).

The software assignable in the package policies are also registered in the LDAP.

# 5.7. Specification of an update point using the package maintenance policy

Feedback

A **Maintenance** policy (see Section 4.6) in the UMC modules for computer and domain management can be used to specify a point at which the following steps should be performed:

- Check for available release updates to be installed (see Section 5.4.4) and, if applicable, installation.
- Installation/uninstallation of package lists (see Section 5.6.5)
- Installation of available errata updates

Alternatively, the updates can also be performed when the system is booting or shut down.

**Table 5.2. 'General' tab**

| Attribute | Description |
|---|---|
| Perform maintenance after system startup | If this option is activated, the update steps are performed when the computer is started up. |
| Perform maintenance before system shutdown | If this option is activated, the update steps are performed when the computer is shut down. |
| Use Cron settings | If this flag is activated, the fields *Month*, *Day of week*, *Day*, *Hour* and *Minute* can be used to specify an exact time when the update steps should be performed. |
| Reboot after maintenance | This option allows you to perform an automatic system restart after release updates either directly or after a specified time period of hours. |

# 5.8. Central monitoring of software installation statuses with the software monitor

Feedback

The software monitor is a database in which information is stored concerning the software packages installed across all UCS systems. This database offers an administrator an overview of which release and package versions are installed in the domain and offers information for the step-by-step updating of a UCS domain and for use in identifying problems.

The software monitor can be installed from the Univention App Center with the application *Software installation monitor*. Alternatively, the software package ***univention-pkgdb*** can be installed. Additional information can be found in Section 5.6.

UCS systems update their entries automatically when software is installed, uninstalled or updated. The system on which the software monitor is operated is located by the DNS service record `_pkgdb._tcp`.

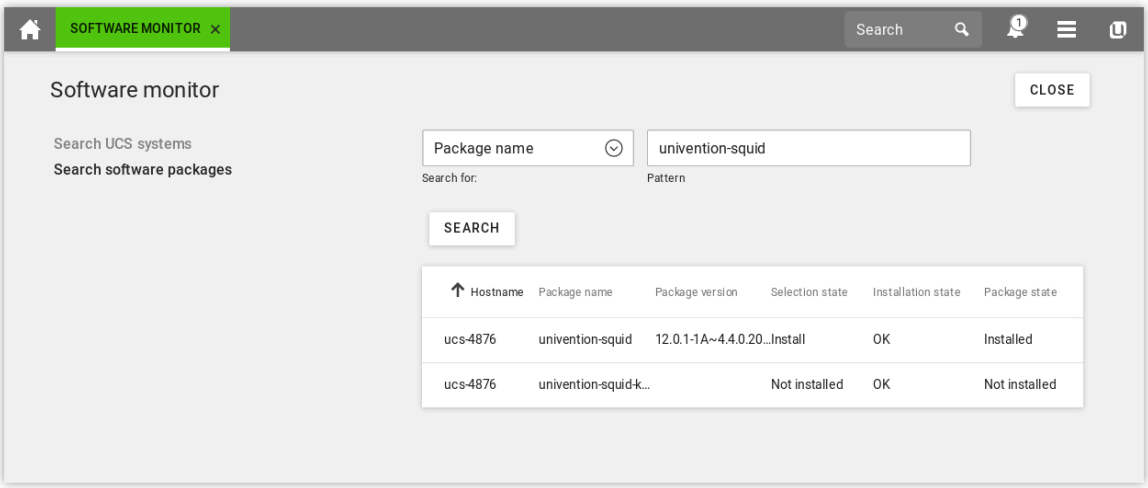
The software monitor's web-based interface integrates in Univention Management Console and can be accessed via the **Software monitor** module. The following functions are available:

- *Systems* allows to search for the version numbers of installed systems. It is possible to search for system names, UCS versions and system roles.
- *Packages* allows to search in the installation data tracked by the package status database. Besides searching for a *Package name* there are various search possibilities available for the installation status of packages:
- The *Selection state* influences the action taken when updating a package. `Install` is used to select a package for installation. If a package is configured to `Hold` it will be excluded from further updates. There

are two possibilities for uninstalling a package: A package removed with `DeInstall` keeps locally created configuration data, whilst a package removed with `Purge` is completely deleted.

- The *Installation state* describes the status of an installed package in relation to upcoming updates. The normal status is `Ok`, which leads to a package being updated when a newer version exists. If a package is configured to `Hold` it will be excluded from the update.
- The *Package state* describes the status of a set-up package. The normal status here is `Installed` for installed packages and `ConfigFiles` for removed packages. All other statuses appear when the package's installation was canceled in different phases.

**Figure 5.9. Searching for packages in the software monitor**




If you do not wish UCS systems to store installation processes in the software monitor (e.g., when there is no network connection to the database), this can be arranged by setting the Univention Configuration Registry variable `pkgdb/scan` to `no`.

Should storing be reactivated at a later date, the command `univention-pkgdb-scan` must be executed to ensure that package versions installed in the meanwhile are also adopted in the database.

The following command can be used to remove a system's program inventory from the database again:

```
univention-pkgdb-scan --remove-system RECHNERNAME
```

# Chapter 6. User management

6.1. User management with Univention Management Console ..... 107
6.2. User activation for apps ..... 114
6.3. User password management ..... 115
6.4. Password settings for Windows clients when using Samba ..... 116
6.5. User self services ..... 117
6.5.1. Password change by user via Univention Management Console ..... 117
6.5.2. Password management via Self Service app ..... 117
6.5.3. Contact information ..... 118
6.5.4. Self registration ..... 120
6.5.4.1. Account creation ..... 121
6.5.4.2. Verification email ..... 122
6.5.4.3. Account verification ..... 123
6.5.5. Self *deregistration* ..... 124
6.6. Automatic lockout of users after failed login attempts ..... 125
6.6.1. Samba Active Directory Service ..... 125
6.6.2. PAM-Stack ..... 126
6.6.3. OpenLDAP ..... 126
6.7. User templates ..... 127
6.8. Overlay module for recording an account's last successful LDAP bind ..... 128

UCS integrates central identity management. All user information are managed centrally in UCS via Univention Management Console and stored in the LDAP directory service.

All the services integrated in the domain access the central account information, i.e., the same username and password are used for the user login to a Windows client as for the login on the IMAP server.

The domain-wide management of user data reduces the administrative efforts as changes do not need to be subsequently configured on different individual systems. Moreover, this also avoids subsequent errors arising from inconsistencies between the individual datasets.

There are three different types of users in UCS:

1. *Normal user account*: Normal user accounts have all available properties. These users can log on to UCS or Windows systems and, depending on the configuration, also to the installed Apps. The users can be administered via the UMC module **Users** (see Section 6.1).

2. *Address book entries*: Address book entries can be used to maintain internal or external contact information. These contacts can not log on to UCS or Windows systems. Address book entries can be managed via the UMC module **Contacts**.

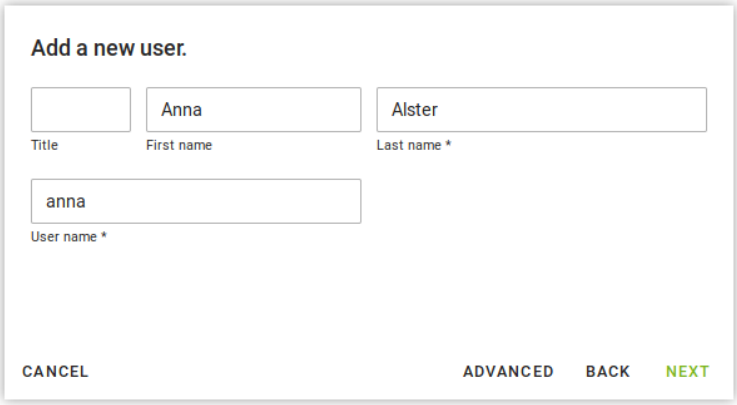
3. *Simple authentication account*: With a simple authentication account, a user object is created, which has only a user name and a password. With this account, only authentication against the LDAP directory service is possible, but no logon to UCS or Windows systems. Simple authentication accounts can be accessed via the UMC module **LDAP directory** (see Section 4.5).

## 6.1. User management with Univention Management Console

Feedback

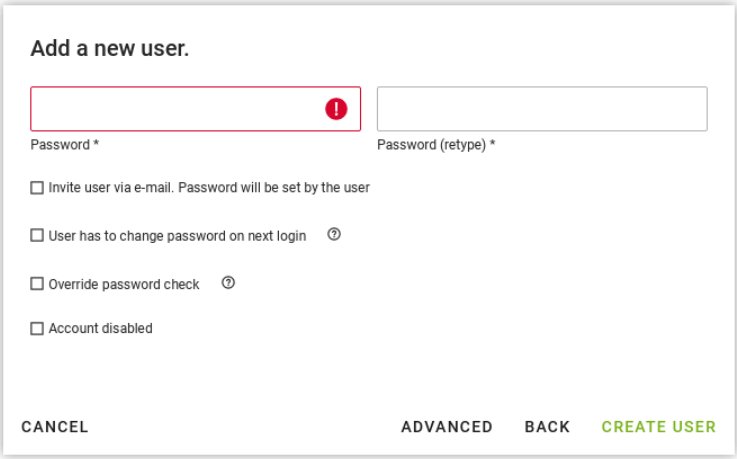
Users are managed in the UMC module *Users* (see Section 4.4).

### Figure 6.1. Creating a user in UMC



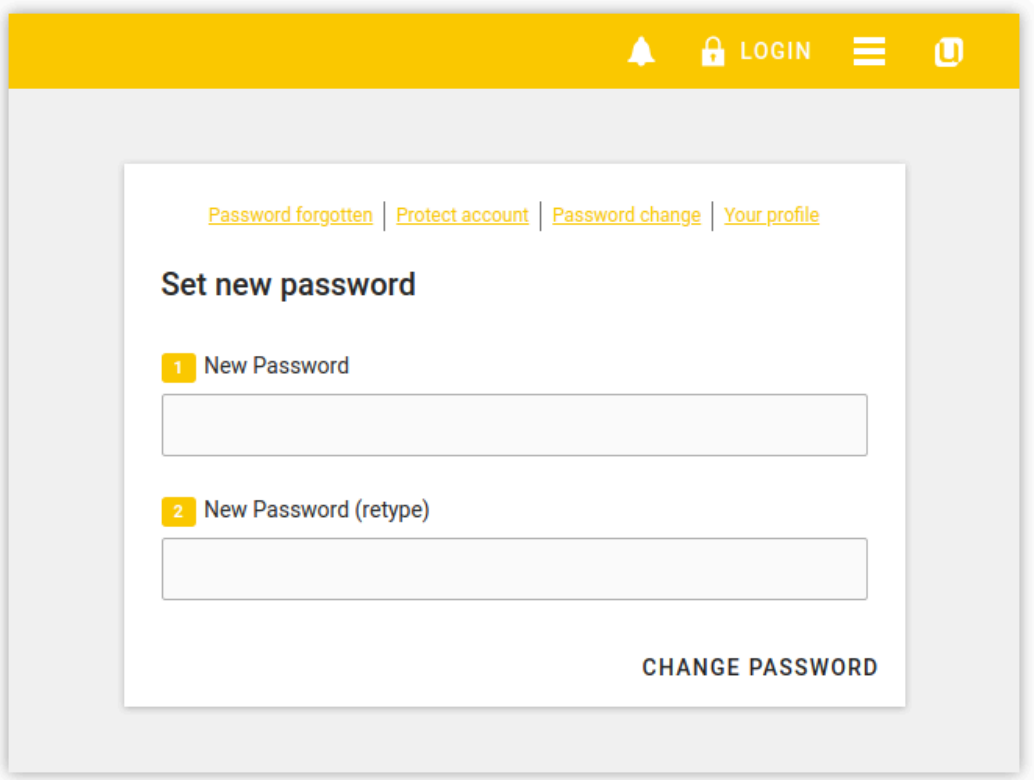

With **Next** on Figure 6.1 the second page Figure 6.2 is shown, where the initial password can be set.

### Figure 6.2. Password setting for a new user




As an alternative the user may set the initial password himself if the Self Service app is installed. For this to work an external e-mail address must be given, which is registered at the contact e-mail address. The user will then get an e-mail to that address containing a web address and a token, which can be used to set the password and unlock the account. For this also see Section 6.5.2.

**Figure 6.3. Initial user password**




In the default setting, a simplified wizard for creating a user is shown, which only requests the most important settings. All attributes can be shown by clicking on **Advanced**. The simplified wizard can be deactivated by setting the Univention Configuration Registry variable `directory/manager/web/modules/users/user/wizard/disabled` to `true`.

**Figure 6.4. Advanced user settings**

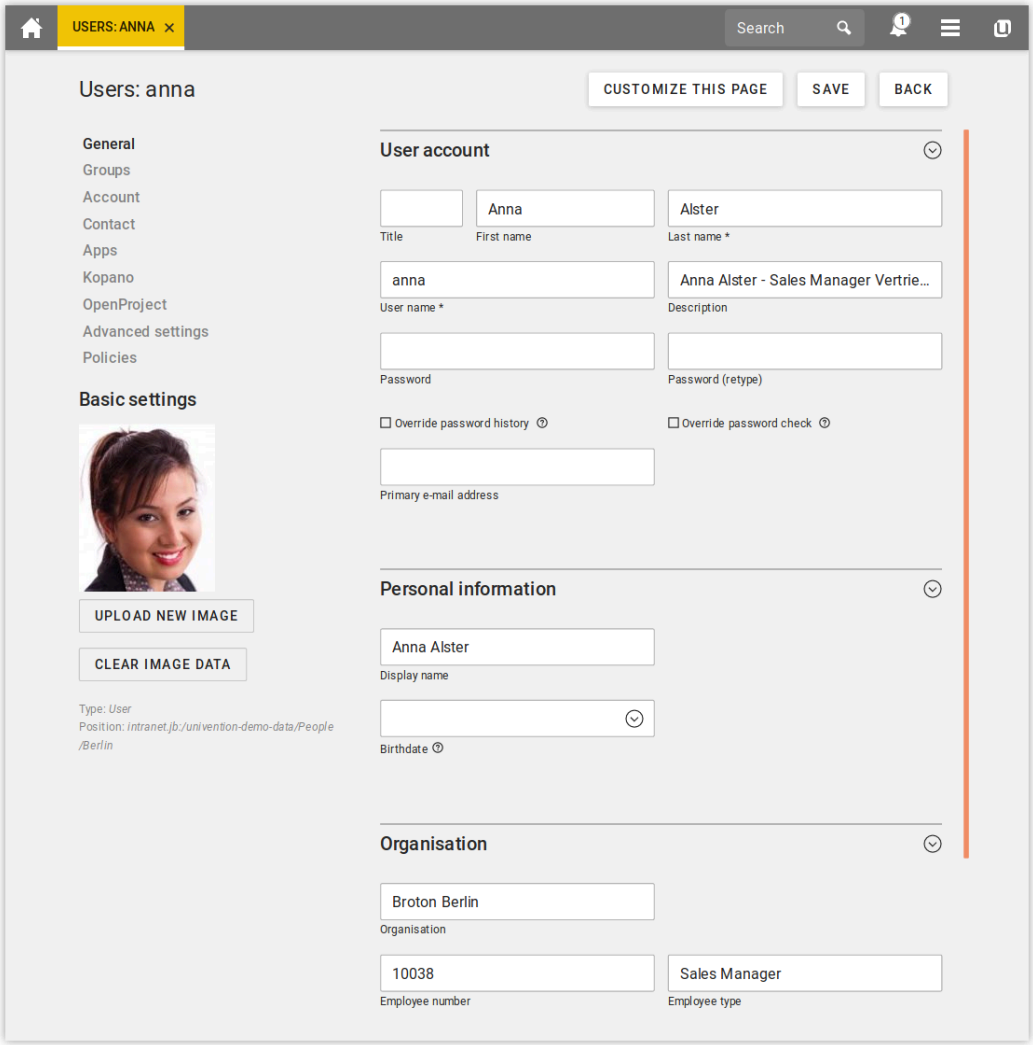



**Table 6.1. 'General' tab**

| Attribute | Description |
|---|---|
| User name | This is the name, by which the user logs into the system. The name has to begin with a letter which has to be followed by: letters a-z in lower case, numerals 0-9, dots, hyphens, or underlines. User names may not contain blank spaces.<br><br>In order to ensure compatibility to non-UCS systems the creation of users which are only distinguished from each other by upper and lower case letters is prevented. Thus, if the user name `smith` already exists, then the user name `Smith` cannot be created.<br><br>In the default setting, it is not possible to create a user with the same name as an existing group. If the Univention Configuration Registry variable `directory/manager/user_group/uniqueness` is set to `false`, this check is removed. |
| Description | Arbitrary descriptions for the user can be entered here. |

| Attribute | Description |
|---|---|
| Password | The user's password has to be entered here. |
| Password (retype) | In order to avoid spelling errors, the user's password has to be entered for a second time. |
| Override password history | By checking this box, the password history is overridden for this user and for this password change. This means, with this change the user can be assigned a password which is already in use. Further details on user password management can be found in Section 6.3. |
| Override password check | By checking this box, the requirements for the length of the password and for password quality checks are overridden for this user and for this password change. This means, the user can e.g. be assigned a shorter password than would be possible according to the defined minimum length. Further details on the password policies for users can be found in Section 6.3. |
| Primary e-mail address | The e-mail address of the user is declared here, see Section 14.3.2. |
| Title | The title of the user is to be entered here. |
| First name | The first name of the user is to be entered here. |
| Last name | The last name of the user is to be entered here. |
| Display name | The display name is automatically composed of the first and surnames. It generally does not need to be changed. The screen name is used for the synchronization with Active Directory and Samba 4 among other things. |
| Organization | The organization is to be entered here. |
| Birthday | This field is used to save a user's birthday. |
| Picture of the user (JPEG format) | This mask can be used to save a picture of the user in LDAP in JPEG format. In the default settings the file size is limited to 512 kilobytes. |
| Employee number | Numbers for staff members can be entered in this field. |
| Employee type | The category of the staff member can be entered here. |
| Superior | The superior of the user can be selected here. |

### Table 6.2. 'Groups' tab

| Attribute | Description |
|---|---|
| Primary group | This select list can be used for specifying the user's primary group. All the groups registered in the domain are open for selection. By default, the group `Domain Users` is preset. |
| Groups | Here it is possible to set further group memberships for the user in addition to the primary group. |

### Table 6.3. 'Account' tab

| Attribute | Description |
|---|---|
| Account is deactivated | The **Account is deactivated** checkbox can be used to deactivate the user account. If this is activated, the user cannot log into the system. This affects all authentication methods. This is typically used when a user leaves the company. In a heterogeneous environment, an account deactivation might also be caused by external tools. |

| Attribute | Description |
|---|---|
| Account expiry date | A date is specified in this input field on which the account will automatically be locked. This is practical for user accounts that only need to be active for a certain period of time, e.g., for interns.<br><br>If the date is deleted or replaced by a different, future date, the user will regain the right to log in. |
| Change password on next login | If this checkbox is ticked, then the user has to change his password during the next login procedure. |
| Password expiry date | If the password is subject to an expiry date, then this date is displayed in this entry field. This entry field cannot be edited directly, see Section 6.3.<br><br>If a password expiry interval is defined, the password expiry date is automatically adjusted when passwords are changed.<br><br>If no **Expiry interval** is declared, the old expiry date will be deleted and no new date will be set. |
| Reset lockout | If the account has automatically been locked temporarily for security reasons, usually because the user has entered the password incorrectly too often, this checkbox can be used to unlock the account again manually before the lockout is lifted automatically when the lockout duration has passed. This temporary account lockout can happen if a corresponding domain wide policy setting has been defined by an administrator. There are three different mechanisms that may trigger lockout if configured properly:<br><br>◦ Failed PAM authentication attempts to an UCS server (see Section 6.6).<br><br>Failed LDAP authentication attempts (if the `ppolicy` overlay has been activated and configured).<br><br>Failed Samba/AD authentication attempts (if the Samba domain *passwordsettings* have been configured). |
| Lockout ends | If the account has automatically been locked temporarily for security reasons, usually because the user has entered the password incorrectly too often, this field shows the time when the account automatically gets unlocked. |
| Activation date | In a user account shall only become usable at a later date, this can be set here. A cron job periodically checks if accounts need to be activated. It runs every 15 minutes by default. When saving the changes, the account is automatically marked as deactivated in case a date in the future has been specified. |
| Windows home drive | If the Windows home directory for this user is to show up on a different Windows drive than that specified by the Samba configuration, then the corresponding drive letter can be entered here, e.g. `M:`. |
| Windows home path | The path of the directory which is to be the user's Windows home directory, is to be entered here, e.g. `\\ucs-file-server\smith` |
| Windows logon script | The user-specific logon script relative to the NETLOGON share is entered here, e.g. `user.bat`. |

| Attribute | Description |
|---|---|
| Windows profile directory | The profile directory for the user can be entered here, e.g. `\\ucs-file-server\user\profile`. |
| Relative ID | The relative ID (RID) is the local part of the SID. If a user is to be assigned a certain RID, the ID in question can be entered in this field. If no RID is assigned, the next available RID will automatically be used. The RID cannot be subsequently changed. Integers from 1000 upwards are permitted. RIDs below 1000 are reserved to standard groups and other special objects. |
| Samba privileges | This selection mask can be used to assign a user selected Windows systems rights, for example the permission to join a system to the domain. |
| Permitted times for Windows logins | This input field contains time periods for which this user can log on to Windows computers.<br><br>If no entry is made in this field, the user can log in at any time of day. |
| Allow the authentication only on these Microsoft Windows hosts | This setting specifies the clients where the user may log in. If no settings are made, the user can log into any client. |
| UNIX home directory | The path of the user's home directory. |
| Login shell | The user's login shell is to be entered in this field. This program is started if the user performs a text-based login. By default, `/bin/bash` is preset. |
| User ID | If the user is to be assigned a certain user ID, the ID in question can be entered in this field. If no value is specified, a free user ID is assigned automatically.<br><br>The user ID can only be declared when adding the user. When the user data are subsequently edited, the user ID will be represented in gray and barred from change. |
| Group ID of the primary group | The group ID of the user's primary group is shown here. The primary group can be changed in the **General** tab. |
| Home share | If a share is selected here, the home directory is stored on the specified server. If no selection is made, the user data are saved on the respective login system. |
| Home share path | The path of the home directory relative to the **Home share** is declared here. The username is already preset as a default value when creating a user. |

### Table 6.4. 'Contact' tab

| Attribute | Description |
|---|---|
| E-mail address(es) | Additional e-mail addresses can be saved here. These are not evaluated by the mail server.<br><br>The values of this attribute are stored in the LDAP attribute *mail*. Most address books applications using an LDAP search function will search for an e-mail address by this attribute. |
| Telephone number(s) | This field contains the user's business phone number. |
| Room number | The room number of the user. |
| Department number | The department number of the user can be entered here. |

| Attribute | Description |
|---|---|
| Street | The street and house number of the user's business address can be entered here. |
| Postal code | This field contains the post code of the user's business address. |
| City | This field contains the city of the user's business address. |
| Private telephone number(s) | The private fixed network phone number can be entered here. |
| Mobile telephone number(s) | The user's mobile numbers can be entered here. |
| Pager telephone number(s) | Pager numbers can be entered here. |
| Private postal address | One or more of the user's private postal addresses can be entered in this field. |

**Table 6.5. 'Mail' tab**

| |
|---|
| This tab is displayed in the advanced settings.<br><br>The settings are described in Section 14.3.2. |

**Table 6.6. '(Options)' tab**

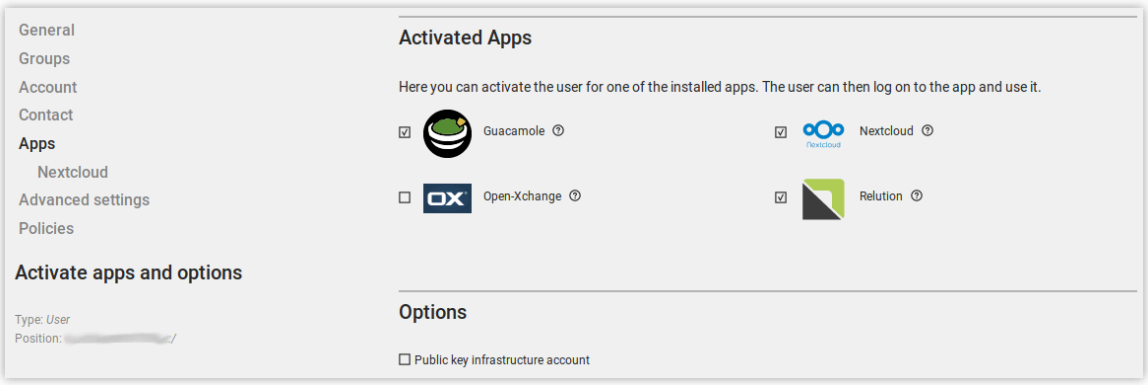
| Attribute | Description |
|---|---|
| Public key infrastructure account | If this checkbox is not ticked, the user will not be assigned the object class `pkiUser`. |

# 6.2. User activation for apps

Feedback

Many apps from the App Center are compatible with the central identity management in UCS. This allows system administrators to activate the users for apps. In some cases, app specific settings for the user can be made. This depends on the app and how it uses the identity management.

**Figure 6.5. User activation for installed apps**




Once an app with user activation is installed in the UCS environment, it will appear with the logo in the **Apps** tab of the user in the UMC module *users*. With a tick in the checkbox the user is activated for the app. If the app offers specific settings another tab with the name of the app will appear to set these parameters. The app activation and the parameters are stored at the user object in the LDAP directory service.

To withdraw a user activation for an app, it is sufficient to deselect the checkbox.

When the app is uninstalled, the checkbox of the user activation for the app is removed from the **Apps** tab of the user in the UMC module.

## 6.3. User password management

Feedback

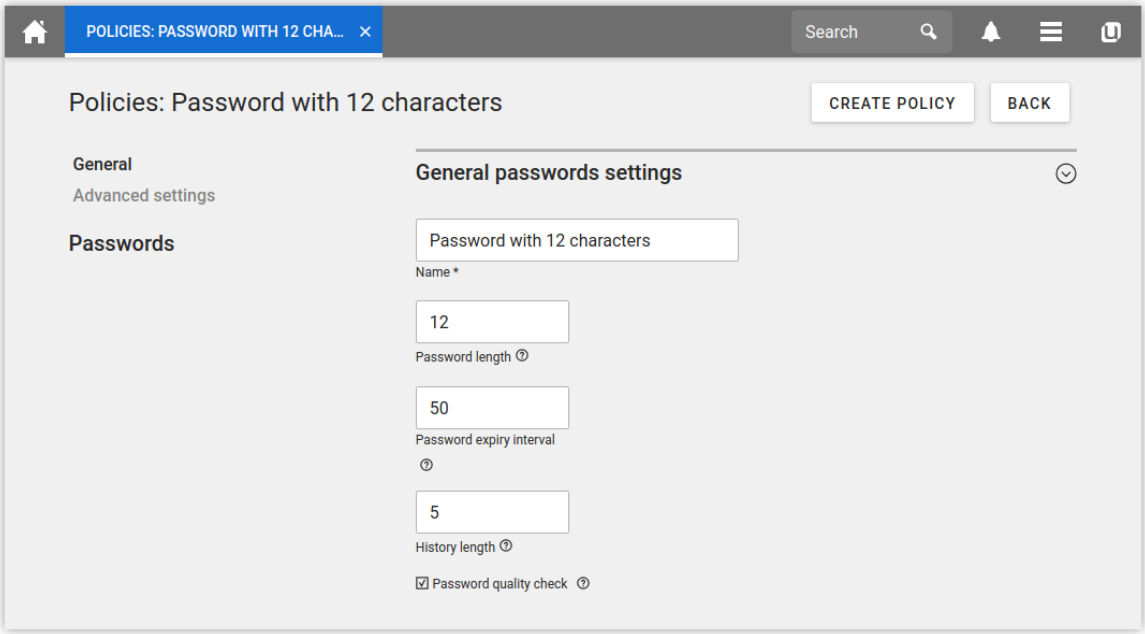
Passwords which are difficult to guess and regular password changes are an essential element of the system security of a UCS domain. The following properties can be configured for users using a *password* policy. If Samba is used, the settings of the Samba domain object (see Section 6.4) apply for logins to Window clients. The settings of the Samba domain object and the policy should be set identically, otherwise different password requirements will apply for logins to Windows and UCS systems.

The password is saved in different attributes for every user saved in the management system:

- The *krb5Key* attribute stores the Kerberos password.
- The *userPassword* attribute stores the Unix password (In other Linux distributions present in `/etc/shadow`).
- The *sambaNTPassword* attribute stores the NT password hash used by Samba.

Password changes are always initiated via Kerberos, either in the UCS PAM configuration or via Samba.

**Figure 6.6. Configuring a password policy**




- The *history length* saves the last password hashes. These passwords can then not be used by the user as a new password when setting a new password. With a password history length of five, for example, five new passwords must be set before a password can be reused. If no password history check should be performed, the value must be set to 0.

  The passwords are not stored retroactively. Example: If ten passwords were stored, and the value is reduced to three, the oldest seven passwords will be deleted during the next password change. If then the value is increased again, the number of stored passwords initially remains at three, and is only increased by each password change.

- The *password length* is the minimum length in characters that a user password must comply with. If no value is entered here, the minimum size is eight characters. The default value of eight characters for password length is fixed, so it always applies if no policy is set and the **Override password check** checkbox is not ticked. This means it even applies if the *default-settings* password policy has been deleted. If no password length check should be performed, the value must be set to 0. A per server default can be configured via Univention Configuration Registry variable `password/quality/length/min`, which applies to users that are not subject to a *UDM* password policy. See the Univention Configuration Registry variable description for details.
- A *password expiry interval* demands regular password changes. A password change is demanded during logons to Univention Management Console, to Kerberos, on Windows clients and on UCS systems following expiry of the period in days. The remaining validity of the password is displayed in the user management under **Password expiry date** in the **Account** tab. If this input field is left blank, no password expiry interval is applied.
- If the option *Password quality check* is activated, additional checks - including dictionary checks - are performed for password changes in Samba, Univention Management Console and Kerberos.

  The configuration is done via Univention Configuration Registry and should occur on all login servers. The following checks can be enforced:

  - Minimum number of digits in the new password (`password/quality/credit/digits`).
  - Minimum number of uppercase letters in the new password (`password/quality/credit/upper`).
  - Minimum number of lowercase letters in the new password (`password/quality/credit/lower`).
  - Minimum number of characters in the new password which are neither letters nor digits (`password/quality/credit/other`).
  - Individual characters/digits can be excluded (`password/quality/forbidden/chars`).
  - Individual characters/figures can be made compulsory (`password/quality/required/chars`).
  - Standard Microsoft password complexity criteria can be applied (`password/quality/mspolicy`). This can be done in addition to the `python-cracklib` checks (value `yes`) or instead of them (`sufficient`). See Univention Configuration Registry variable description for details.

## 6.4. Password settings for Windows clients when using Samba

Feedback

With the Samba domain object, one can set the password requirements for logins to Windows clients in a Samba domain.

The Samba domain object is managed via the UMC module *LDAP directory*. It can be found in the `samba` container below the LDAP base and carries the domain's NetBIOS name.

The settings of the Samba domain object and the policy (see Section 6.3) should be set identically, otherwise different password requirements will apply for logins to Windows and UCS systems.

### Table 6.7. 'General' tab

| Attribute | Description |
|---|---|
| Password length | The minimum number of characters for a user password. |

| Attribute | Description |
|---|---|
| Password history | The latest password changes are saved in the form of hashes. These passwords can then not be used by the user as a new password when setting a new password. With a password history of five, for example, five new passwords must be set before a password can be reused. |
| Minimum password age | The period of time set for this must have at least expired since the last password change before a user can reset his password again. |
| Maximum password age | Once the saved period of time has elapsed, the password must be changed again by the user the next time he logs in. If the value is left blank, the password is infinitely valid. |

# 6.5. User self services

Feedback

## 6.5.1. Password change by user via Univention Management Console

Feedback

In Univention Management Console, every user can reset his password via the **Change password** module. The module can also be opened by selecting the **Settings -> Change password** entry in the top right user menu. To change the password, the current password must be entered first. The change is performed directly via the PAM stack (see Section 8.4.4) and is then available centrally for all services.

## 6.5.2. Password management via Self Service app

Feedback

By installing the UCS components **Self Service Backend** and **Self Service** in the domain via the **App Center**, users are enabled to take care of their password management without administrator interaction.

The Self Service app registers one web service on the portal which can be accessed via a dedicated web page: "Change Password". It allows users to update their password given their old password as well as to reset their lost password by requesting a token to be sent to a previously registered contact e-mail address. The token has to be entered on the dedicated password reset web page.

The following Univention Configuration Registry variables can be used to activate or deactivate individual features of the password management.

`umc/self-service/passwordreset/frontend/enabled`

Activates or deactivates the "Password forgotten" page of the Self Service on individual hosts.

`umc/self-service/passwordreset/backend/enabled`

Activates or deactivates the backend of the "Password forgotten" page. This Univention Configuration Registry variable has to be set on the systems that is defined as Self Service backend via the Univention Configuration Registry variable `self-service/backend-server`, since requests regarding these variables are forwarded to the Self Service backend.

`umc/self-service/protect-account/frontend/enabled`

Activates or deactivates the "Protect account" page of the Self Service on individual hosts.

`umc/self-service/protect-account/backend/enabled`

Activates or deactivates the backend of the "Protect account" page. This Univention Configuration Registry variable has to be set on the systems that is defined as Self Service backend via the Univention

Configuration Registry variable `self-service/backend-server`, since requests regarding these variables are forwarded to the Self Service backend.

`umc/self-service/passwordchange/frontend/enabled`

Activates or deactivates the "Password change" page of the Self Service on individual hosts.

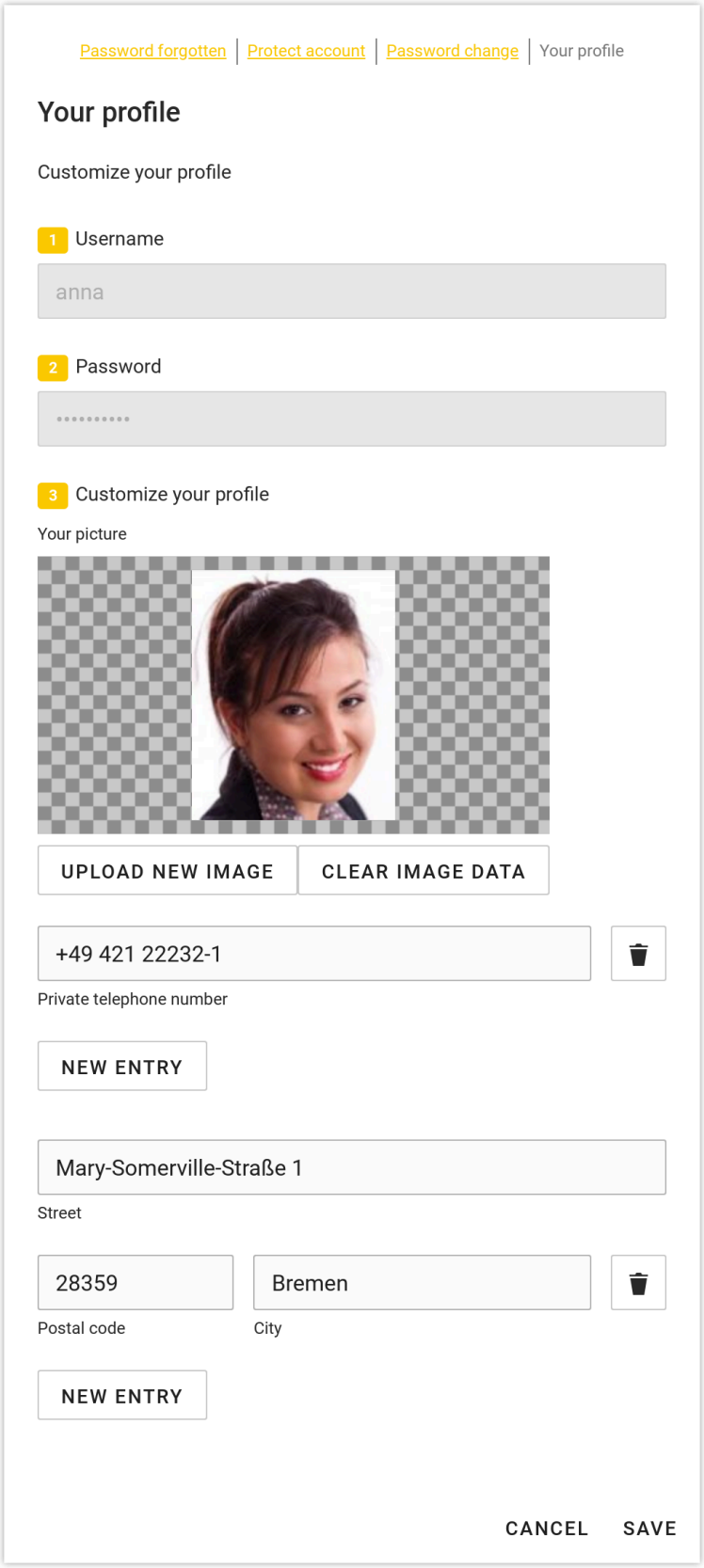
The design of the Self Service app can be customized via the file `/usr/share/univention-self-service/www/css/custom.css`. This file is not overwritten during an update.

## 6.5.3. Contact information

Feedback

Additional personal data can be stored in LDAP with the users account. This may include a picture, the users private address and other contact information. By default only administrators can modify them. As an alternative selected attributes may be unlocked for the user to change himself. The user then can do this using the Self Service app.

**Figure 6.7. User profile self-service**



For this the following Univention Configuration Registry variables must be configured:[1]

`self-service/ldap_attributes`

This variable configured the *LDAP* attributes a user can modify at its own user account. The names of the attributes must be separated by comma. This variable must be set on master domain controller (and backup domain controllers).

`self-service/udm_attributes`

This variable configured the *UDM* attributes a user can modify. The names of the attributes must be separated by comma. This variable must be set on the hosts, where the Self Service app is installed (incl. master domain controller).

`umc/self-service/profiledata/enabled`

This variable must be set to `true` on all involved server systems to enable the mechanism.

`umc/self-service/allow-authenticated-use`

This variable defines whether the specification of user name and password is necessary when opening and modifying your own user profile if you are already logged in to Univention Portal. As of UCS 4.4-7, this Univention Configuration Registry variable is automatically set to `true` when the *Self Service* is installed for the first time. The `true` value activates the use of an existing Univention Portal session. The fields **Username** and **Password** are then automatically filled in or no longer displayed. Systems upgraded to UCS 4.4-7 will retain the old behavior by automatically setting the value to `false`. Note that this variable must be set to the same value on all participating systems where the Self Service app is installed (incl. master domain controller).

## 6.5.4. Self registration

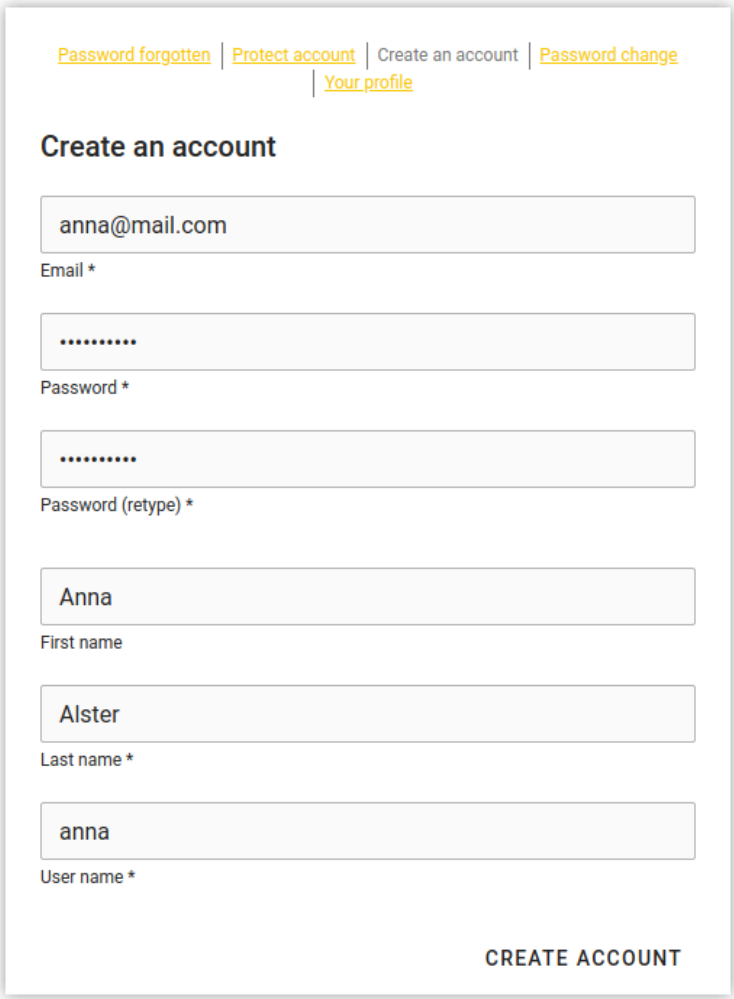
Feedback

The Self Service allows for users to register themselves, which will create a user account that has to be verified via email. User accounts that are created via the Self Service will have the *RegisteredThroughSelfService* attribute of the user set to TRUE and the *PasswordRecoveryEmailVerified* attribute set to FALSE. After the user has verified his account via the verification email the *PasswordRecoveryEmailVerified* attribute will be set to TRUE.

[1]Both attributes variables must match each other. The names of the attributes and their mapping can be fetch from the following command: `python -c 'from univention.admin.handlers.users.user import mapping;print("\n".join(map("{0[0]:>30s} {0[1][0]:<30s}".format, sorted(mapping._map.items()))))'`

#### 6.5.4.1. Account creation

**Figure 6.8. Account registration**




The "Create an account" page can be enabled/disabled on individual hosts, where the Self Service app is installed, with the Univention Configuration Registry variable `umc/self-service/account-registration/frontend/enabled`.

Aspects about the "Create an account" page and the account creation itself can be configured with the following Univention Configuration Registry variables. These Univention Configuration Registry variables have to be set on the systems that is defined as Self Service backend via the Univention Configuration Registry variable `self-service/backend-server`, since requests regarding these variables are forwarded to the Self Service backend.

`umc/self-service/account-registration/backend/enabled`

With this variable the account registration can be disabled/enabled.

`umc/self-service/account-registration/usertemplate`

With this variable a user template (Section 6.7) can be specified that will be used for the creation of self registered accounts.

`umc/self-service/account-registration/usercontainer`

With this variable a container can be specified under which the self registered users are created.

`umc/self-service/account-registration/udm_attributes`

This variable configures which UDM attributes of a user account are shown on the "Create an account" page of the Self Service. The names of the UDM attributes must be provided as a comma separated list.

`umc/self-service/account-registration/udm_attributes/required`

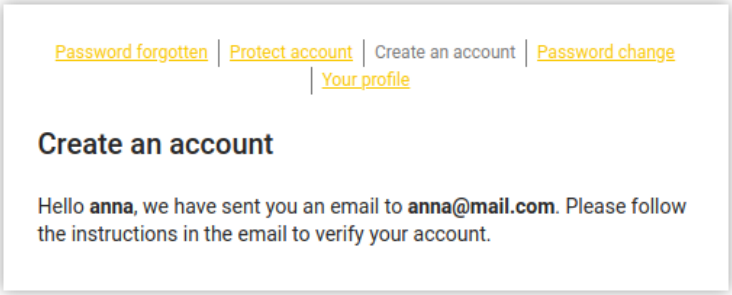
This variable configures which of the UDM attributes set via the Univention Configuration Registry variable `umc/self-service/account-registration/udm_attributes` are required for the user to provide. The names of the UDM attributes must be provided as a comma separated list.

### 6.5.4.2. Verification email

Feedback

After a user has clicked on **Create account**, they will see a message that an email for the account verification has been sent.

**Figure 6.9. Sending the verification email**




Aspects about the verification email and the verification token can be configured with the following Univention Configuration Registry variables. These Univention Configuration Registry variables have to be set on the Self Service backend that is defined via the Univention Configuration Registry variable `self-service/backend-server`, since requests regarding these variables are forwarded to the Self Service backend.

`umc/self-service/account-verification/email/webserver_address`

Defines the *host* part to use in the verification link URL. The default is to use the FQDN of the Self Service backend defined via the Univention Configuration Registry variable `self-service/backend-server` since this Univention Configuration Registry variable is evaluated there.

`umc/self-service/account-verification/email/sender_address`

Defines the sender address of the verification email. Default is `Account Verification Service <noreply@FQDN>`.

`umc/self-service/account-verification/email/server`

Server name or IP address of the mail server to use.

`umc/self-service/account-verification/email/text_file`

A path to a text file whose content will be used for the body of the verification email. The text can contain the following strings which will be sub-

stituted accordingly: *{link}*, *{token}*, *{tokenlink}* and *{username}*. As default the file `/usr/lib/python2.7/dist-packages/univention/management/console/modules/passwordreset/sending/verification_email_body.txt` is used.

`umc/self-service/account-verification/email/token_length`

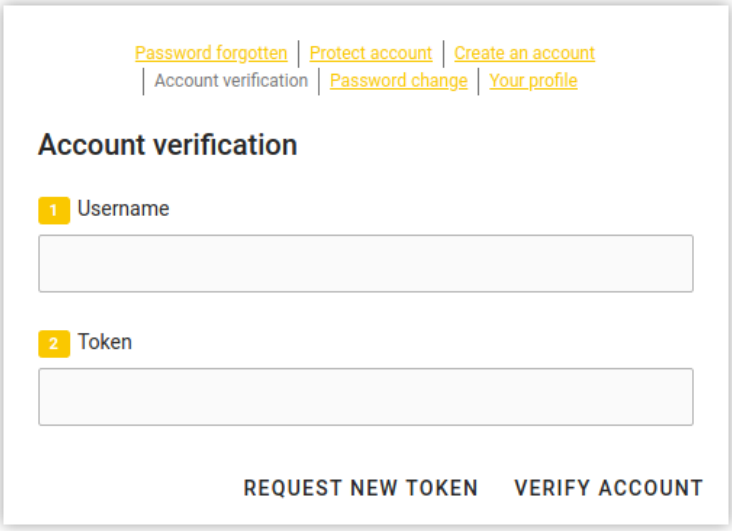
Defines the number of characters that is used for the verification token. Defaults to 64.

### 6.5.4.3. Account verification

Feedback

Following the verification link from the email, the user will land on the "Account verification" page of the Self Service.

**Figure 6.10. Account verification**

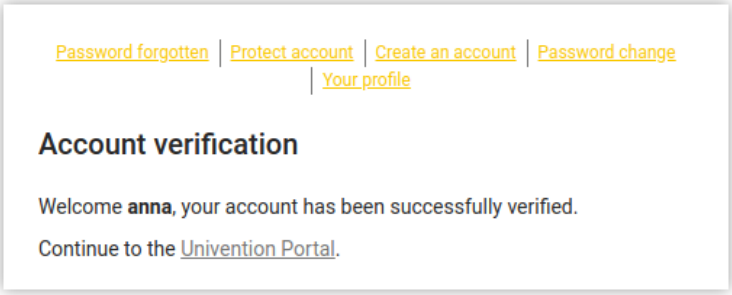



The "Account verification" page can be enabled/disabled on individual hosts, where the Self Service app is installed, with the Univention Configuration Registry variable `umc/self-service/account-verification/frontend/enabled`.

The account verification and request of new verification tokens can be disabled/enabled with the Univention Configuration Registry variable `umc/self-service/account-verification/backend/enabled`. This Univention Configuration Registry variable has to be set on the systems that is defined as Self Service backend via the Univention Configuration Registry variable `self-service/backend-server`.

After the verification was successful a message is shown on what the user can do next. This message can be configured with the Univention Configuration Registry variable `umc/self-service/account-verification/next-steps`. A localized message can be configured by appending a locale to the variable. E.g. `umc/self-service/account-verification/next-steps/en`.

**Figure 6.11. Account verification**




The SSO login can be configured to deny login from unverified, self registered accounts. This is configured through the Univention Configuration Registry variable `saml/idp/selfservice/check_email_verification`. This needs to be set on the domain controller master and all domain controller backups. The setting has no effect on accounts created by an administrator.

The message on the SSO login page for unverified, self registered accounts, can be set with the Univention Configuration Registry variables `saml/idp/selfservice/account-verification/error-title` and `saml/idp/selfservice/account-verification/error-descr`. A localized message can be configured by adding a local to the variable. E.g. `saml/idp/selfservice/account-verification/error-title/en`.

## 6.5.5. Self *deregistration*

Feedback

The Self Service allows for users to request the deletion of their own account. This feature can be activated with the Univention Configuration Registry variable `umc/self-service/account-deregistration/enabled`, which will show a "Delete my account" Button on the "Your profile" page of the Self Service (Section 6.7).

If a user has requested to delete his account, it will not be deleted directly but deactivated. In addition the *DeregisteredThroughSelfService* attribute of the user will be set to `TRUE` and the *DeregistrationTimestamp* attribute of the user will be set to the current time in the *GeneralizedTime* LDAP syntax[2]. If the user has a *PasswordRecoveryEmail* set they will receive a notification email which can be configured with the following Univention Configuration Registry variables.

`umc/self-service/account-deregistration/email/sender_address`

Defines the sender address of the email. Default is `Password Reset Service <noreply@FQDN>`.

`umc/self-service/account-deregistration/email/server`

Server name or IP address of the mail server to use.

`umc/self-service/account-deregistration/email/text_file`

A path to a text file whose content will be used for the body of the email. The text can contain the following strings which will be substituted accordingly: *`{username}`*. As default the file `/usr/lib/python2.7/dist-packages/univention/management/console/modules/passwordreset/sending/deregistration_notification_email_body.txt` is used.

The Self Service provides a script under `/usr/share/univention-self-service/delete_deregistered_accounts.py` that can be used to delete all users/user objects which

[2] https://ldapwiki.com/wiki/GeneralizedTime

have *DeregisteredThroughSelfService* set to TRUE and whose *DeregistrationTimestamp* is older than a specified time. The following command would delete users whose *DeregistrationTimestamp* is older than 5 days and 2 hours.

```
/usr/share/univention-self-service/delete_deregistered_accounts.py \
 --timedelta-days 5 --timedelta-hours 2
```

For all possible arguments to the script see:

```
/usr/share/univention-self-service/delete_deregistered_accounts.py \
 --help
```

The script can be run regularly by creating a cron job via Univention Configuration Registry.

```
ucr set cron/delete_deregistered_accounts/command=\
/usr/share/univention-self-service/delete_deregistered_accounts.py\
' --timedelta-days 30' \
 cron/delete_deregistered_accounts/time='00 06 * * *'  # daily at 06:00
```

More information on how to set cron jobs via Univention Configuration Registry can be found in Section 8.4.8.3.

# 6.6. Automatic lockout of users after failed login attempts

Feedback

By default, a user can enter her password incorrectly any number of times. To hinder brute force attacks on passwords, an automatic lockout for user accounts can be activated after a configured number of failed log-in attempts.

UCS unifies various methods for user authentication and authorization. Depending on the installed software components, there may be different mechanisms for configuring and counting failed logon attempts.

The three different methods are described below.

## 6.6.1. Samba Active Directory Service

Feedback

In Samba Active Directory environments, various services are provided by Samba, such as Kerberos. To lockout users after too many failed log-in attempts, the tool `samba-tool` can be used.

- `samba-tool domain passwordsettings show` shows the currently configured values.
- `samba-tool domain passwordsettings set --account-lockout-threshold=5` specifies how often a user can attempt to log in with an incorrect password before the account is locked.
- `samba-tool domain passwordsettings set --account-lockout-duration=3` specifies the number of minutes an account will be locked after too many incorrect passwords have been entered.
- `samba-tool domain passwordsettings set --reset-account-lockout-after=5` defines the number of minutes after which the counter is reset. If an account gets automatically unlocked after the lockout duration, the counter is not reset immediately, to keep the account under strict monitoring for some time. During the time window between the end of the lockout duration and the point when the the counter gets reset, a single attempt to log in with an incorrect password will lock the account immediately again.

The manual unlocking of a user is done in the user administration on the tab **Account** by activating the checkbox **Unlock account**.

## 6.6.2. PAM-Stack

Feedback

The automatic locking of users after failed logons in the PAM stack can be enabled by setting the Univention Configuration Registry variable `auth/faillog` to `yes`. The upper limit of failed log-in attempts at which an account lockout is configured in the Univention Configuration Registry variable `auth/faillog/limit`. The counter is reset each time the password is entered correctly.

The lockout is activated locally per system by default. In other words, if a user enters her password incorrectly too many times on one system, she can still login on another system. Setting the Univention Configuration Registry variable `auth/faillog/lock_global` will make the lock effective globally and register it in the LDAP directory. The global lock can only be set on master domain controller/Backup systems as other system roles do not have the necessary permissions in the LDAP directory. On all systems with any of these system roles, the lockout gets automatically activated locally or deactivated again via the listener module, depending on the current lock state in the LDAP directory.

As standard, the lockout is not subject to time limitations and must be reset by the administrator. However, it can also be reset automatically after a certain time interval has elapsed. This is done by specifying a time period in seconds in the Univention Configuration Registry variable `auth/faillog/unlock_time`. If the value is set to 0, the lock is reset immediately.

By default, the `root` user is excluded from the password lock, but can also be subjected to it by setting the Univention Configuration Registry variable `auth/faillog/root` to `yes`.

If accounts are only locked locally, the administrator can unlock a user account by entering the command `faillog -r -u USERNAME`. If the lock occurs globally in the LDAP directory, the user can be reset in Univention Management Console on the tab **Account** in the user options **Unlock account**.

## 6.6.3. OpenLDAP

Feedback

On domain controllers, automatic account locking can be enabled for too many failed LDAP server logon attempts. The MDB LDAP backend must be used. This is the default backend since UCS 4, previous systems must be migrated to the MDB LDAP backend, see [ucs-performance-guide].

Automatic account locking must be enabled for each domain controller. To do this, the Univention Configuration Registry variables `ldap/ppolicy` and `ldap/ppolicy/enabled` must be set to `yes` and the OpenLDAP server must be restarted:

```
ucr set ldap/ppolicy=yes ldap/ppolicy/enabled=yes
/etc/init.d/slapd restart
```

The default policy is designed so that five repeated failed LDAP server logon attempts within five minutes cause the lockout. A locked account can only be unlocked by a domain administrator via Univention Management Console.

The number of repeated failed LDAP server logon attempts can be adjusted in the configuration object with the `objectClass` `pwdPolicy`:

```
univention-ldapsearch objectclass=pwdPolicy
```

The `pwdMaxFailure` attribute determines the number of LDAP authentication errors before locking. The `pwdMaxFailureCountInterval` attribute determines the time interval in seconds that is considered. Failed logon attempts outside this interval are ignored in the count.

The following command can be used to block the account after 10 attempts:

```
ldapmodify -x -D cn=admin,$(ucr get ldap/base) -y /etc/ldap.secret
 <<__EOT__
```

```
dn: cn=default,cn=ppolicy,cn=univention,$(ucr get ldap/base)
changetype: modify
replace: pwdMaxFailure
pwdMaxFailure: 10
__EOT__
```

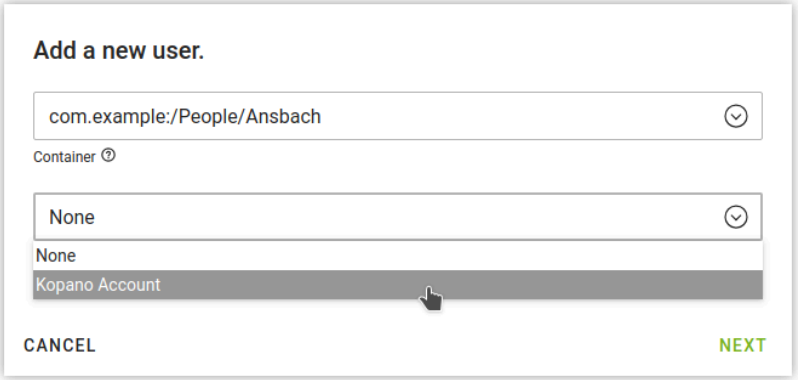
The manual unlocking of a user is done in the user administration on the tab **Account** by activating the checkbox **Unlock account**.

## 6.7. User templates

Feedback

A user template can be used to preset settings when creating a user. If at least one user template is defined, it can be selected when creating a user.

**Figure 6.12. Selecting a user template**




User templates are administrated in the UMC module **LDAP directory**. There one needs to switch to the `univention` container and then to the `templates` subcontainer. A new user template can be created here via the **Add** with the object type **Settings: User template**.

In a user template, either a fixed value can be specified (e.g., for the address) or an attribute of the user management referenced. Attributes are then referenced in chevrons.

A list of possible attributes can be displayed with the command:

```
univention-director-manager users/user
```

in the section *users/user variables* of the output.

If a user template is used for adding a user, this template will overwrite all the fields with the preset values of the template. In doing so, an empty field is set to "".

It is also possible to only use partial values of attributes or convert values in uppercase/lowercase.

For example, the UNIX home directory can be stored under `/home/<title>.<lastname>` or the primary e-mail address can be predefined with **`<firstname>.<lastname>@company.com`**. Substitutions are generally possibly for any value, but there is no syntax or semantics check. So, if no first name is specified when creating a user, the above e-mail address would begin with a dot and would thus be invalid according to the e-mail standard. Similar sources of error can also occur when handling file paths etc. Non-resolvable attributes (for instance due to typing errors in the template) are deleted.

If only a single character of an attribute is required instead of the complete attribute value, the index of the required character can be entered in the user template in square parentheses after the name of the attribute.

The count of characters of the attribute begins with 0, so that index 1 corresponds to the second character of the attribute value. Accordingly, **<firstname>[0].<lastname>@company.com** means an e-mail address will consist of the first letter of the first name plus the lastname.

A substring of the attribute value can be defined by entering a range in square parentheses. In doing so, the index of the first required character and the index of the last required character plus one are to be entered. For example, the input **<firstname>[2:5]** returns the third to fifth character of the first name.

Adding `:lower` or `:upper` to the attribute name converts the attribute value to lowercase or uppercase, e.g., **<firstname:lower>**. If a modifier like `:lower` is appended to the entire field, the complete value is transformed, e.g. **<lastname>@company.com<:lower>**.

The option `:umlauts` can be used to convert special characters such as è, ä or ß into the corresponding ASCII characters.

The option `:alphanum` can be used to remove all non alphanumeric characters such as ` or #. A whitelist of characters that are ignored by this option can be defined in the UCR variable `directory/manager/templates/alphanum/whitelist`. If this option is applied to an entire field, even manually placed symbols like the @ in an email address are removed. To avoid that, this option should be applied to specific attributes only or desired symbols should be entered into the whitelist.

The options `:strip` or `:trim` remove all white space characters from the start and end of the string.

It is also possible to combine options, e.g: `:umlauts,upper`.

# 6.8. Overlay module for recording an account's last successful LDAP bind

Feedback

## Caution

Before using this feature please read this support article about activating the OpenLDAP `lastbind` overlay module[3].

The optional `lastbind` overlay module[4] for OpenLDAP allows recording the timestamp of the last successful LDAP bind in the *authTimestamp* attribute and can for example be used to detect unused accounts.

The `lastbind` overlay module can be activated by setting the Univention Configuration Registry variable `ldap/overlay/lastbind` to `yes` and restarting the OpenLDAP server. When the module is activated on an UCS server, a timestamp is written to the account's *authTimestamp* attribute when that account logs into the LDAP server. The Univention Configuration Registry variable `ldap/overlay/lastbind/precision` can be used to configure the time in seconds that has to pass before the *authTimestamp* attribute is updated. This prevents a large number of write operations that can impair performance.

The *authTimestamp* attribute can only be queried on the LDAP server where the `lastbind` overlay module is activated. It is not replicated to other LDAP servers. For that reason the `/usr/share/univention-ldap/univention_lastbind.py` script can be executed to collect the youngest *authTimestamp* value from all reachable LDAP servers in the UCS domain and save it into the *lastbind* extended UDM attribute of a user. The script can be invoked to update the *lastbind* extended attribute of one or all users. The *lastbind* extended attribute maps to the *univentionAuthTimestamp* LDAP attribute.

One way to keep the *lastbind* extended attribute up-to-date is by creating a cron job via UCR:

[3] https://help.univention.com/t/14404
[4] http://manpages.ubuntu.com/manpages/xenial/man5/slapo-lastbind.5.html

```
ucr set cron/update_lastbind_attribute/command='/usr/share/univention-
ldap/univention_lastbind.py --allusers' \
       cron/update_lastbind_attribute/time='00 06 * * *'  # daily at
 06:00 a.m.
```

More information on how to set cron jobs via UCR can be found in Section 8.4.8.3.

# Chapter 7. Group management

7.1. Managing groups in Univention Management Console ..... 131
7.2. Nested groups ..... 134
7.3. Local group cache ..... 134
7.4. Synchronization of Active Directory groups when using Samba 4 ..... 135
7.5. Overlay module for displaying the group information on user objects ..... 135

Permissions in UCS are predominantly differentiated between on the basis of *groups*. Groups are stored in the LDAP and are thus identical on all systems. Groups can contain not only user accounts, but can also optionally accept computer accounts.

In addition, there are also local user groups on each system, which are predominantly used for hardware access. These are not managed through the UCS management system, but saved in the `/etc/group` file.

The assignment of users to groups is performed in two ways:

- A selection of groups can be assigned to a user in the user management (see Section 6.1)
- A selection of users can be assigned to a group in the group management (see Section 7.1)

## 7.1. Managing groups in Univention Management Console

Feedback

Groups are managed in the UMC module *Groups* (see Section 4.4).

**Figure 7.1. Creating a group in UMC**

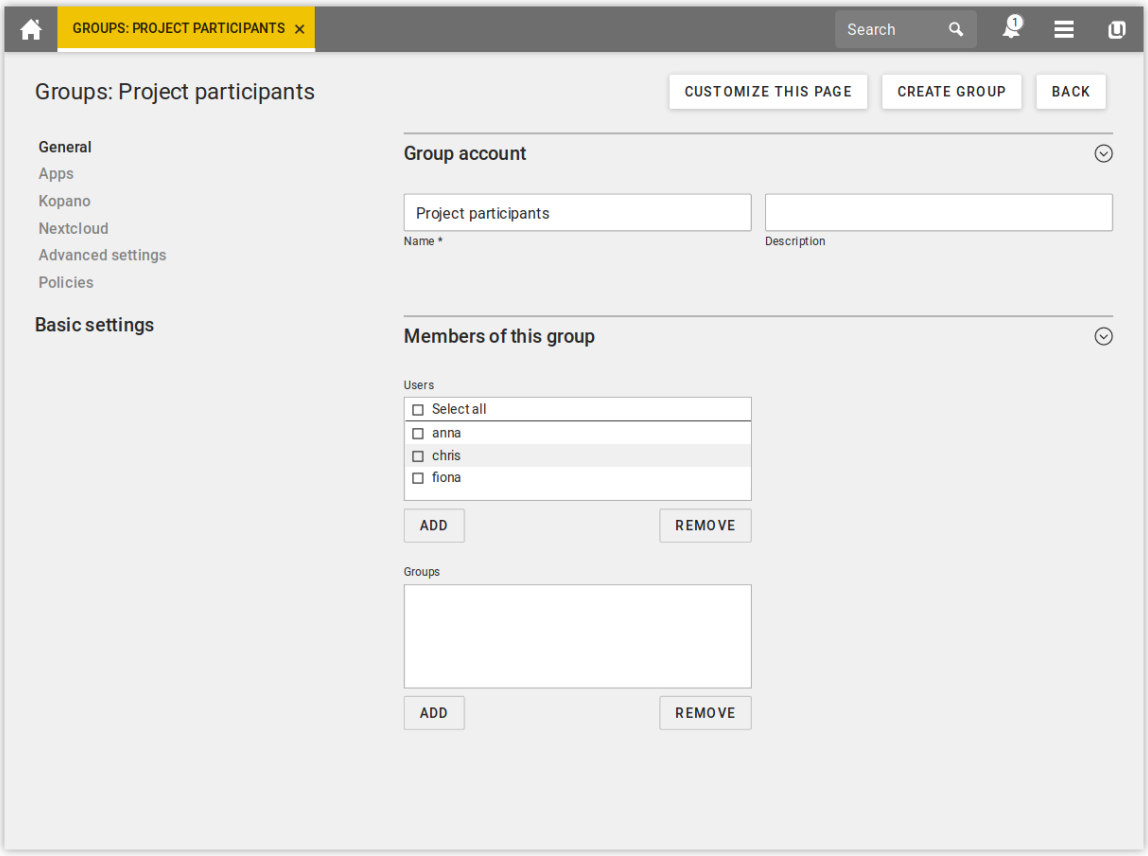



**Table 7.1. 'General' tab**

| Attribute | Description |
|---|---|
| Name (*) | The name of the group has to begin and end with a letter or a numeral. The rest of the characters which form the group name may include letters, numerals, spaces, hyphens, or dots.<br><br>In the default setting, it is not possible to create a group with the same name as an existing user. If the Univention Configuration Registry variable `directory/manager/user_group/uniqueness` is set to `false`, this check is removed. |
| Description | A description of the group can be entered here. |
| Users | This input field can be used for adding users as members to the group. |
| Groups | On this input field, other groups can be added as members of the current group (groups in groups). |

**Table 7.2. 'Advanced settings' tab**

| Attribute | Description |
|---|---|
| Mail | These options define a mail group and are documented in the Section 14.3.4. |
| Host members | This field can be used for accepting computers as members of the group. |

| Attribute | Description |
|---|---|
| Nested groups | The current group can be added as a member to other groups here (groups in groups). |
| Group ID | If a group is to be assigned a certain group ID, the ID in question can be entered in this field. Otherwise, Univention Management Console will automatically assign the next available group ID when adding the group. The group ID cannot be subsequently changed. When editing the group, the group ID will be represented in gray.<br><br>The group ID may consist of integers between 1000 and 59999 and between 65536 and 100000. |
| Windows -> Relative ID | The relative ID (RID) is the local part of the Security ID (SID) and is used in Windows and Samba domains. If a group is to be assigned a certain RID, the ID in question can be entered in this field. Otherwise, Univention Management Console will automatically assign the next available group ID when adding the group.<br><br>The RID cannot be subsequently changed. When editing the group, the group ID will be represented in gray.<br><br>The RIDs below 1000 are reserved for standard groups and other special objects.<br><br>When Samba 4 is used, the RID is generated by Samba and cannot be specified. |
| Windows -> group type | This group type is evaluated when the user logs on to a Samba/AD-based domain Three types of Windows groups can be distinguished:<br><br>◦ *Domain Groups* are known across the domain. This is the default group time.<br><br>◦ *Local groups* are only relevant on Windows servers. If a local group is created on a Windows server, this group is known solely to the server; it is not available across the domain. UCS, in contrast, does not differentiate between local and global groups. After taking over an AD domain, local groups in UCS can be handled in the same way as global groups.<br><br>◦ *Well-known group*: This group type covers groups preconfigured by Samba/Windows servers which generally have special privileges, e.g., `Power Users`. |
| Windows -> AD group type | This group type is only evaluated when the user logs on to a Samba 4-based domain (which offers Active Directory domain services). These groups are described in Section 7.4. |
| Windows -> Samba privileges | This input mask can be used to assign Windows system rights to a group, e.g., the right to join a Windows client in the domain. This function is documented in Section 6.1. |

**Table 7.3. 'Options' tab**

| **This tab is only available when adding groups, not when editing groups. Certain LDAP object classes for the group can be de-selected here. The entry fields for the attributes of these classes can then no longer be filled in.** | |
|---|---|
| **Attribute** | **Description** |
| Samba group | This checkbox indicates whether the group contains the object class `sambaGroupMapping`. |
| POSIX group | This checkbox indicates whether the group contains the object class `posixGroup`. |

# 7.2. Nested groups

Feedback

UCS supports group nesting (also known as "groups in groups"). This simplifies the management of the groups. For example, if two locations are managed in one domain, two groups can be formed `IT staff location A` and `IT staff location B`), to which the user accounts of the location's IT staff can be assigned respectively.

To create a cross-location group, it is then sufficient to define the groups `IT staff location A` and `IT staff location B` as members.

Cyclic dependencies of nested groups are automatically detected and refused. This check can be disabled with the Univention Configuration Registry variable `directory/manager/web/modules/groups/group/checks/circular_dependency`. Cyclic memberships must also be avoided in direct group changes without the UCS management system.

The resolution of nested group memberships is performed during the generation of the group cache (see Section 7.3) and is thus transparent for applications.

# 7.3. Local group cache

Feedback

The user and computer information retrieved from the LDAP is cached by the Name Server Cache Daemon, see Section 8.4.9.

Since UCS 3.1, the groups are no longer cached via the NSCD for performance and stability reasons; instead they are now cached by the NSS module ***libnss-extrausers***. The group information is automatically exported to the `/var/lib/extrausers/group` file by the `/usr/lib/univention-pam/ldap-group-to-file.py` script and read from there by the NSS module.

In the basic setting, the export is performed every 15 minutes by a cron job and is additionally started if the Univention Directory Listener has been inactive for 15 seconds. The interval for the cron update is configured in Cron syntax (see Section 8.4.8.2) by the Univention Configuration Registry variable `nss/group/cachefile/invalidate_interval`. This listener module can be activated/deactivated via the Univention Configuration Registry variable `nss/group/invalidate_cache_on_changes` (`true`/`false`).

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

## 7.4. Synchronization of Active Directory groups when using Samba 4

Feedback

If Samba 4 is used, the group memberships are synchronized between the Samba 4 directory service and the OpenLDAP directory service by the Univention S4 connector, i.e., each group on the UCS side is associated with a group in Active Directory. General information on the Univention S4 connector can be found in Section 9.2.2.4.

Some exceptions are formed by the *pseudo groups* (sometimes also called system groups). These are only managed internally by Active Directory/Samba 4, e.g., the `Authenticated Users` group includes a list of all the users currently logged on to the system. Pseudo groups are stored in the UCS directory service, but they are not synchronized by the Univention S4 connector and should usually not be edited. This applies to the following groups:

- `Anonymous Logon`, `Authenticated Users`, `Batch`, `Creator Group`
- `Creator Owner`, `Dialup`, `Digest Authentication`
- `Enterprise Domain Controllers`, `Everyone`, `IUSR`, `Interactive`
- `Local Service`, `NTLM Authentication`, `Network Service`, `Network`
- `Nobody`, `Null Authority`, `Other Organization`, `Owner Rights`
- `Proxy`, `Remote Interactive Logon`, `Restricted`, `SChannel Authentication`
- `Self`, `Service`, `System`, `Terminal Server User`, `This Organization`
- `World Authority`

In Samba 4 / Active Directory, a distinction is made between the following four AD group types. These group types can be applied to two types of groups; *security groups* configure permissions (corresponding to the UCS groups), whilst *distribution groups* are used for mailing lists:

- *Local* groups only exist locally on a host. A local group created in Samba 4 is synchronized by the Univention S4 Connector and thus also appears in the UMC. There is no need to create local groups in the UMC.
- *Global* groups are the standard type for newly created groups in the UMC. A global group applies for one domain, but it can also accept members from other domains. If there is a trust relationship with a domain, the groups there are displayed and permissions can be assigned. However, the current version of UCS does not support multiple domains/forests or outgoing trust relationships.
- *Domain local* groups can also adopt members of other domains (insofar as there is a trust relationship in place or they form part of a forest). Local domain groups are only shown in their own domain though. However, the current version of UCS does not support multiple domains/forests or outgoing trust relationships.
- *Universal* groups can adopt members from all domains and these members are also shown in all the domains of a forest. These groups are stored in a separate segment of the directory service, the so-called global catalog. Domain forests are currently not supported by Samba 4.

## 7.5. Overlay module for displaying the group information on user objects

Feedback

In the UCS directory service, group membership properties are only saved in the group objects and not in the respective user objects. However, some applications expect group membership properties at the user objects

(e.g., in the attribute *memberOf*). An optional overlay module in the LDAP server makes it possible to present these attributes automatically based on the group information. The additional attributes are not written to the LDAP, but displayed on the fly by the overlay module if a user object is queried.

### Caution

Before using this feature please read SDB 1278 about activating the OpenLDAP `memberOf` overlay module.

To this end, the ***univention-ldap-overlay-memberof*** package must be installed on all LDAP servers. Afterwards `/usr/share/univention-ldap-overlay-memberof/univention-update-memberof` must be invoked on all servers.

In the default setting, the user attribute *memberOf* is shown. The Univention Configuration Registry variable `ldap/overlay/memberof/memberof` can be used to configure a different attribute.

http://sdb.univention.de/1278

univention
be open.

# Chapter 8. Computer management


8.1. Management of computer accounts in Univention Management Console ..... 138
8.1.1. Integration of Ubuntu clients ..... 142
8.2. Configuration of hardware and drivers ..... 142
8.2.1. Available kernel variants ..... 142
8.2.2. Hardware drivers / kernel modules ..... 143
8.2.3. GRUB boot manager ..... 143
8.2.4. Network configuration ..... 145
8.2.4.1. Network interfaces ..... 145
8.2.4.2. Configuring proxy access ..... 149
8.2.5. Configuration of the monitor settings ..... 149
8.2.6. Mounting NFS shares ..... 150
8.2.7. Collection of list of supported hardware ..... 150
8.3. Administration of local system configuration with Univention Configuration Registry ..... 151
8.3.1. Introduction ..... 151
8.3.2. Using the Univention Management Console web interface ..... 152
8.3.3. Using the command line front end ..... 152
8.3.3.1. Querying a UCR variable ..... 152
8.3.4. Policy-based configuration of UCR variables ..... 154
8.3.5. Modifying UCR templates ..... 155
8.3.5.1. Referencing of UCR variables in templates ..... 155
8.3.5.2. Integration of inline Python code in templates ..... 155
8.4. Basic system services ..... 156
8.4.1. Administrative access with the root account ..... 156
8.4.2. Configuration of language and keyboard settings ..... 156
8.4.3. Starting/stopping system services / configuration of automatic startup ..... 157
8.4.4. Authentication / PAM ..... 158
8.4.4.1. Limiting authentication to selected users ..... 158
8.4.5. Configuration of the LDAP server in use ..... 159
8.4.6. Configuration of the print server in use ..... 159
8.4.7. Logging/retrieval of system messages and system status ..... 159
8.4.7.1. Log files ..... 159
8.4.7.2. Logging the system status ..... 159
8.4.7.3. Querying system statistics in Univention Management Console ..... 160
8.4.7.4. Process overview in Univention Management Console ..... 160
8.4.7.5. System error diagnosis in Univention Management Console ..... 161
8.4.8. Executing recurring actions with Cron ..... 161
8.4.8.1. Hourly/daily/weekly/monthly execution of scripts ..... 161
8.4.8.2. Defining local cron jobs in `/etc/cron.d/` ..... 161
8.4.8.3. Defining cron jobs in Univention Configuration Registry ..... 162
8.4.9. Name service cache daemon ..... 162
8.4.10. RDP login to systems using XRDP ..... 163
8.4.10.1. Installation ..... 163
8.4.10.2. Configuration ..... 163
8.4.10.3. Client software ..... 164
8.4.10.4. Known issue: Wrong keyboard layout ..... 164
8.4.10.5. Alternatives ..... 164
8.4.11. SSH login to systems ..... 164
8.4.12. Configuring the time zone / time synchronization ..... 165

# 8.1. Management of computer accounts in Univention Management Console

Feedback

All UCS, Linux and Windows systems within a UCS domain each have a computer domain account (also referred to as the host account) with which the systems can authenticate themselves among each other and with which they can access the LDAP directory.

The computer account is generally created automatically when the system joins the UCS domain (see Section 3.2); however, the computer account can also be added prior to the domain join.

The password for the computer account is generated automatically during the domain join and saved in the `/etc/machine.secret` file. In the default setting, the password consists of 20 characters (can be configured via the Univention Configuration Registry variable `machine/password/length`). The password is regenerated automatically at fixed intervals (default setting: 21 days; can be configured using the Univention Configuration Registry variable `server/password/interval`). Password rotation can also be disabled using the variable `server/password/change`.

There is an different computer object type for every system role. Further information on the individual system roles can be found in Section 3.3.

Computer accounts are managed in the UMC module **Computers**.

In the default setting, a simplified wizard for creating a computer is shown, which only requests the most important settings. All attributes can be shown by clicking on **Advanced**. If there is a DNS forward zone and/or a DNS reverse zone (see Section 11.2) assigned to the selected network object (see Section 11.1), a host record and/or pointer record is automatically created for the host. If there is a DHCP service configured for the network object and a MAC address is configured, a DHCP host entry is created (see Section 11.3).

The simplified wizard can be disabled for all system roles by setting the Univention Configuration Registry variable `directory/manager/web/modules/computers/computer/wizard/disabled` to `true`.

**Figure 8.1. Creating a computer in UMC**

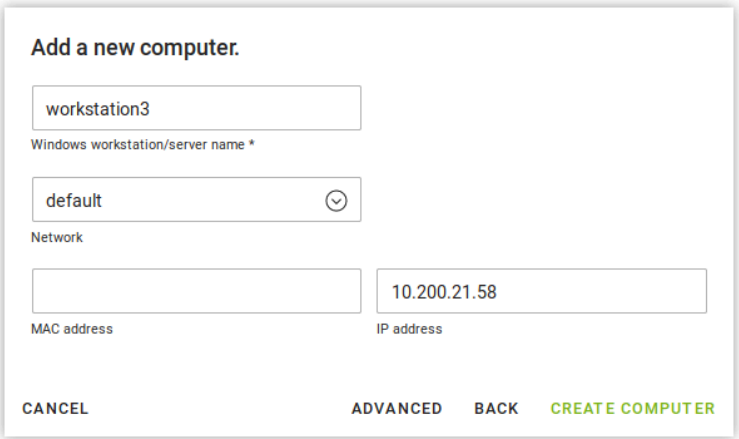


**Figure 8.2. Advanced computer settings**

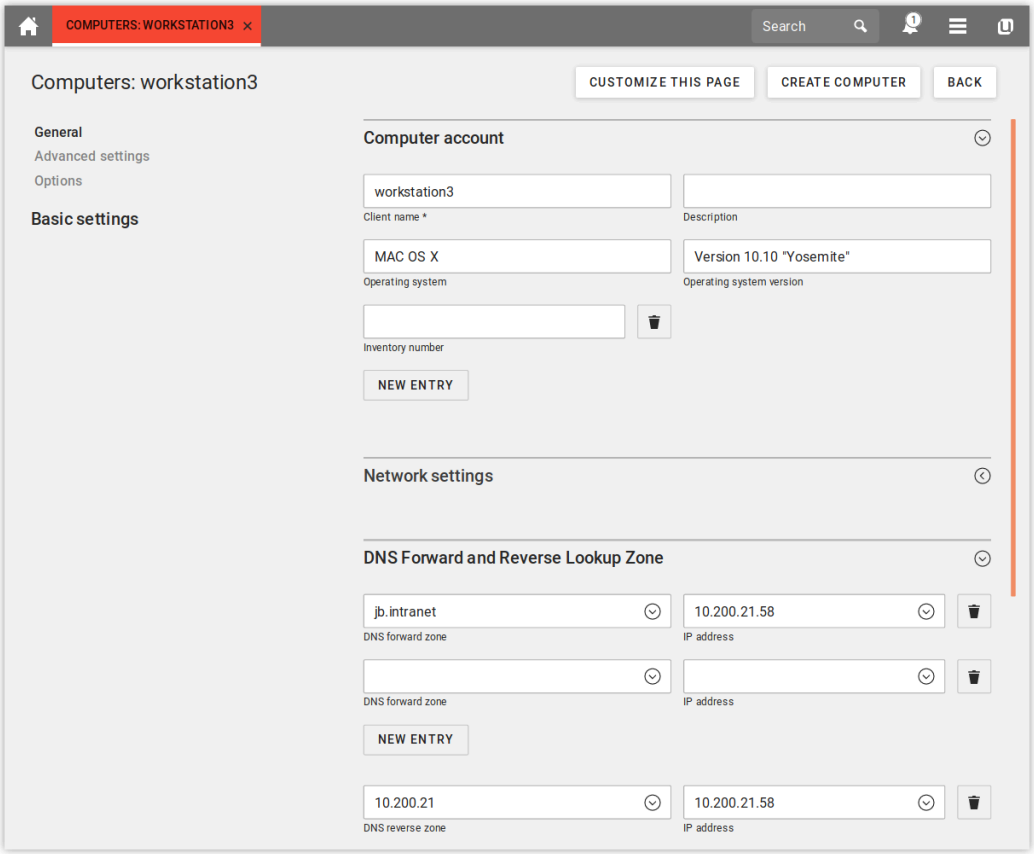



**Table 8.1. 'General' tab**

| Attribute | Description |
|---|---|
| Name | The name for the host should be entered in this input field.<br><br>To guarantee compatibility with different operating systems and services, computer names should only contain the lowercase letters *a* to *z*, numbers, hyphens and underscores. Umlauts and special characters are not permitted. The full stop is used as a separating mark between the individual components of a fully qualified domain name and must therefore not appear as part of the computer name. Computer names must begin with a letter.<br><br>Microsoft Windows accepts computer names with a maximum of 13 characters, so as a rule computer names should be limited to 13 characters if there is any chance that Microsoft Windows will be used.<br><br>After creation, the computer name can only be changed for the system roles *Windows Workstation/Server*, *Mac OS X Client* and *IP managed client*. |
| Description | Any description can be entered for the host in this input field. |
| Inventory number | Inventory numbers for hosts can be stored here. |
| Network | The host can be assigned to a existing network object. Information on the IP configuration can be found in Section 11.1. |

| Attribute | Description |
|---|---|
| MAC address | The MAC address of the computer can be entered here e.g., `2e:44:56:3f:12:32`. If the computer is to receive a DHCP entry, the entry of the MAC address is essential. |
| IP address | Fixed IP addresses for the host can be given here. Further information on the IP configuration can be found in Section 11.1.<br><br>If a network was selected on the **General** tab, the IP address assigned to the host from the network will be shown here automatically.<br><br>An IP address entered here (i.e. in the LDAP directory) can only be transferred to the host via DHCP. If no DHCP is being used, the IP address must be configured locally, see Section 8.2.4.<br><br>If the IP addresses entered for a host are changed without the DNS zones being changed, they are automatically changed in the computer object and - where they exist - in the DNS entries of the forward and reverse lookup zones. If the IP address of the host was entered at other places as well, these entries must be changed manually! For example, if the IP address was given in a DHCP boot policy instead of the name of the boot server, this IP address will need to be changed manually by editing the policy. |
| Forward zone for DNS entry | The DNS forward zone in which the computer is entered. The zone is used for the resolution of the computer name in the assigned IP address. Further information on the IP configuration can be found in Section 11.1. |
| Reverse zone for DNS entry | The DNS reverse zone in which the computer is entered. The zone is used to resolve the computer's IP address in a computer name. Further information on the IP configuration can be found in Section 11.1. |
| DHCP service | If a computer is supposed to procure its IP address via DHCP, a DHCP service must be assigned here. Information on the IP configuration can be found in Section 11.1.<br><br>During assignment, it must be ensured that the DHCP servers of the DHCP service object are responsible for the physical network.<br><br>If a network is selected on the **General** tab an appropriate entry for the network will be added automatically. It can be adapted subsequently. |

**Table 8.2. 'Account' tab' (advanced settings)**

| Attribute | Description |
|---|---|
| Password | The password for the computer account is usually automatically created and rotated. For special cases such as the integration of external systems it can also be explicitly configured in this field.<br><br>The same password must then also be entered locally on the computer in the `/etc/machine.secret` file. |
| Primary group | The primary group of the host can be selected in this selection field. This is only necessary when they deviate from the automatically created default values. The default value for a DC master or DC backup is `DC Backup Hosts`, for a DC slave `DC Slave Hosts` and for member servers `Computers`. |

**Table 8.3. 'Unix account' tab (advanced settings)**

| Attribute | Description |
|---|---|
| Unix home directory (*) | A different input field for the host account can be entered here. The automatically created default value for the home directory is `/dev/null` |
| Login shell | If a different login shell from the default value is to be used for the computer account, the login shell can be adapted manually in this input field. The automatically set default value assumes a login shell of `/bin/sh`. |

**Table 8.4. 'Services' tab (advanced settings)**

| Attribute | Description |
|---|---|
| Service | By means of a service object, applications or services can determine whether a service is available on a computer or generally in the domain. |

## Note

The tab 'Services' is only displayed on UCS server system roles.

**Table 8.5. 'Deployment' tab (advanced settings)**

| |
|---|
| This tab is used for the Univention Net Installer, see [ext-doc-inst]. |

**Table 8.6. 'DNS alias' tab (advanced settings)**

| Attribute | Description |
|---|---|
| Zone for DNS Alias | If a zone entry for forward mapping has been set up for the host in the **Forward zone for DNS entry** field, the additional alias entries via which the host can be reached can be configured here. |

**Table 8.7. 'Groups' tab (advanced settings)**

| |
|---|
| The computer can be added into different groups in this tab. |

**Table 8.8. 'Nagios services' tab (advanced settings)**

| |
|---|
| This tab is used to specify which Nagios tests should be performed for this computer, see Section 15.2.3.3. |

**Table 8.9. 'Nagios notification' tab (advanced settings)**

| |
|---|
| This tab is used to specify which users should be informed if Nagios tests should fail, see Section 15.2.3.3. |

**Table 8.10. 'UVMM' tab (advanced settings)**

| |
|---|
| This tab is used to specify which virtualization servers can be managed by UVMM. Further information can be found in Chapter 16. |

**Table 8.11. '(Options)' tab**

| Attribute | Description |
|---|---|
| This tab allows to disable LDAP object classes for host objects. The entry fields for attributes of disables object classes are no longer shown. Not all object classes can be modified subsequently. | |

| Attribute | Description |
|---|---|
| Kerberos principal | If this checkbox is not selected the host does not receive the `krb5Principal` and `krb5KDCEntry` object classes. |
| POSIX account | If this checkbox is not selected the host does not receive the `posixAccount` object class. |
| Nagios support | If this checkbox is selected Nagios checks can be activated for this host. |
| Samba account | If this checkbox is not selected the host does not receive the `sambaSamAccount` object class. |

### 8.1.1. Integration of Ubuntu clients

Feedback

Ubuntu clients can be managed in Univention Management Console with their own system role. The network properties for DNS/DHCP can also be managed via Univention Management Console.

The use of policies is not supported.

Some configuration adjustments need to be performed on Ubuntu systems; these are documented in the extended documentation [ext-doc-domain].

## 8.2. Configuration of hardware and drivers

Feedback

### 8.2.1. Available kernel variants

Feedback

The standard kernel in UCS 4.4 is based on the Linux kernel 4.9. In principle, there are three different types of kernel packages:

- A *kernel image package* provides an executable kernel which can be installed and started.
- A *kernel source package* provides the source code for a kernel. From this source, a tailor-made kernel can be created, and functions can be activated or deactivated.
- A *kernel header package* provides interface information which is required by external packages if these have to access kernel functions. This information is usually necessary for compiling external kernel drivers.

Normally, the operation of a UCS system only requires the installation of one kernel image package.

The default kernel in UCS for i386-based systems is the so-called *bigmem kernel* for processors with PAE support, which supports 64 GB RAM. For older i386-based systems a second kernel without PAE support is provided, which only supports up to 4 GB RAM. The standard kernel for amd64 systems has no such limits.

Several kernel versions can be installed in parallel. This makes sure that there is always an older version available to which can be reverted in case of an error. So-called meta packages are available which always refer to the kernel version currently recommended for UCS. In case of an update, the new kernel version will be installed, making it possible to keep the system up to date at any time.

The following meta packages are available under i386 / 32 bit:

- ***univention-kernel-image*** - Standard kernel with support up to 64 GB RAM
- ***univention-kernel-image-486*** - Kernel for systems without PAE support (max. 4 GB RAM)

The following meta packages are available under amd64 / 64 bit:

- ***univention-kernel-image*** - Standard kernel

## 8.2.2. Hardware drivers / kernel modules

Feedback

The boot process occurs in two steps using an initial ramdisk ('initrd' for short). This is composed of an archive with further drivers and programs.

The GRUB boot manager (see Section 8.2.3) loads the kernel and the initrd into the system memory, where the initrd archive is extracted and mounted as a temporary root file system. The real root file system is then mounted from this, before the temporary archive is removed and the system start implemented.

The drivers to be used are recognized automatically during system start and loaded via the udev device manager. At this point, the necessary device links are also created under `/dev/`. If drivers are not recognized (which can occur if no respective hardware IDs are registered or hardware is employed which cannot be recognized automatically, e.g., ISA boards), kernel modules to be loaded can be added via Univention Configuration Registry variable `kernel/modules`. If more than one kernel module is to be loaded, these must be separated by a semicolon. The Univention Configuration Registry variable `kernel/blacklist` can be used to configure a list of one or more kernel modules for which automatic loading should be prevented. Multiple entries must also be separated by a semicolon.

Unlike other operating systems, the Linux kernel (with very few exceptions) provides all drivers for hardware components from one source. For this reason, it is not normally necessary to install drivers from external sources subsequently.

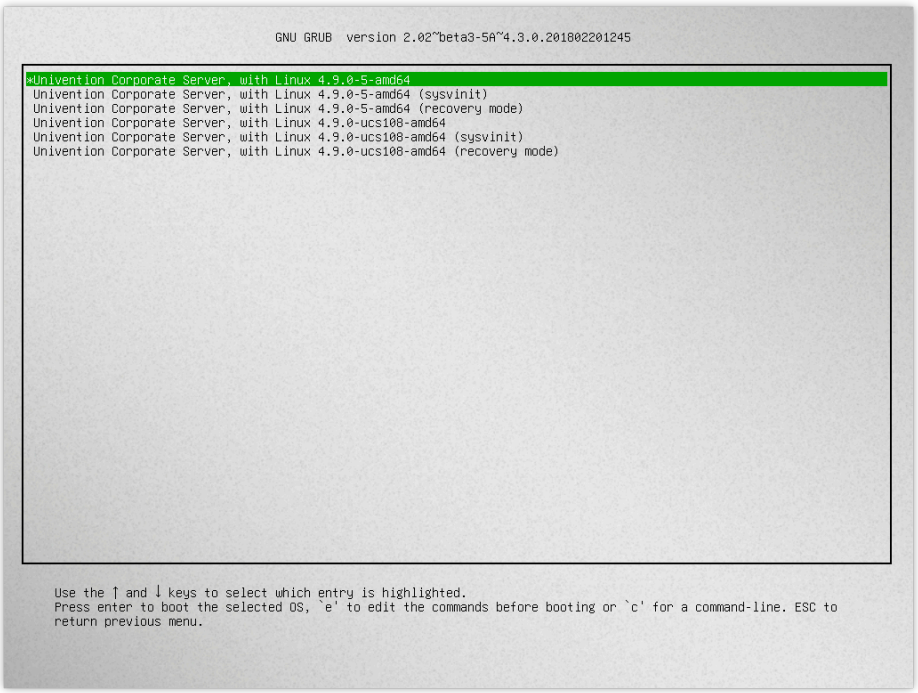
However, if external drivers / kernel modules are required, they can be integrated via the DKMS framework (Dynamic Kernel Module Support). This provides a standardized interface for kernel sources, which are then built automatically for every installed kernel (insofar as the source package is compatible with the respective kernel). For this to happen, the kernel header package ***univention-kernel-headers*** must be installed in addition to the ***dkms*** package. Please note that not all the external kernel modules are compatible with all kernels.

## 8.2.3. GRUB boot manager

Feedback

In Univention Corporate Server GNU GRUB 2 is used as the boot manager. GRUB provides a menu which allows the selection of a Linux kernel or another operating system to be booted. GRUB can also access file systems directly and can thus, for example, load another kernel in case of an error.

**Figure 8.3. GRUB menu**




GRUB gets loaded in a two-step procedure; in the Master Boot Record of the hard drive, the Stage 1 loader is written which refers to the data of Stage 2, which in turn manages the rest of the boot procedure.

The selection of kernels to be started in the boot menu is stored in the file `/boot/grub/grub.cfg`. This file is generated automatically; all installed kernel packages are available for selection. The memory test program `Memtest86+` can be started by selecting the option **Memory test** and performs a consistency check for the main memory.

There is a five second waiting period during which the kernel to be booted can be selected. This delay can be changed via the Univention Configuration Registry variable `grub/timeout`.

By default a screen of *800x600* pixels size and 16 Bit color depth is pre-set. A different value can be set via the Univention Configuration Registry variable `grub/gfxmode`. Only resolutions are supported which can be set via VESA BIOS extensions. A list of available modes can be found at https://en.wikipedia.org/wiki/VESA_BIOS_Extensions. The input must be specified in the format ***HORIZONTALxVERTICAL@COLOURDEPTHBIT***, so for example `1024x768@16`.

Kernel options for the started Linux kernel can be passed with the Univention Configuration Registry variable `grub/append`. Univention Configuration Registry variable `grub/xenhopt` can be used to pass options to the Xen hypervisor.

The graphic representation of the boot procedure - the so-called splash screen - can be deactivated by setting Univention Configuration Registry variable `grub/bootsplash` to `nosplash`.

Older Xen environments might use a version of PyGrub, which still requires the GRUB 1 configuration file `/goot/grub/menu.lst` to boot paravirtualized Xen systems. This file is generated automatically if it does not yet exist. This behavior can be deactivated by setting the Univention Configuration Registry variable `grub/generate-menu-lst` to `no`.

## 8.2.4. Network configuration

Feedback

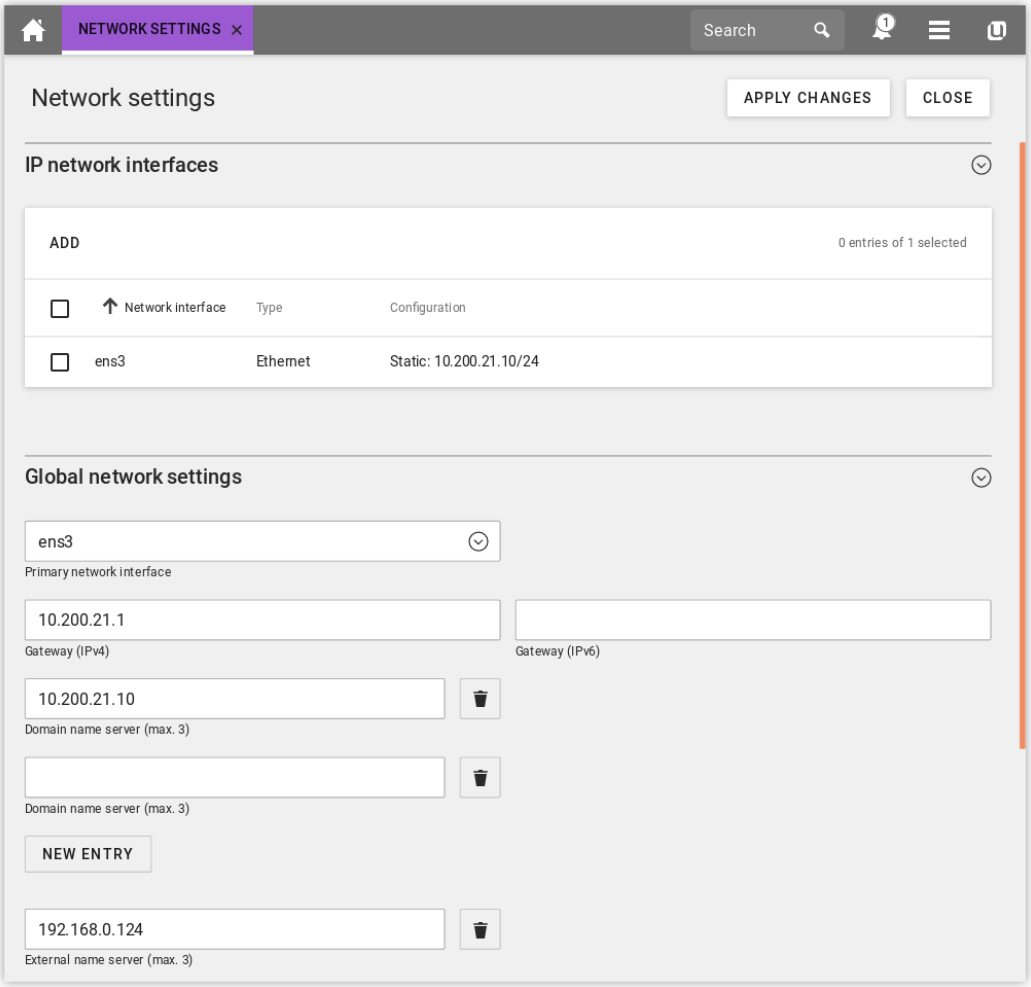
### 8.2.4.1. Network interfaces

Feedback

The configuration of network interfaces can be adjusted in Univention Management Console via the module **Network settings**.

The configuration is saved in Univention Configuration Registry variables, which can also be set directly. These variables are listed in parentheses in the individual sections.

**Figure 8.4. Configuring the network settings**




All the network cards available in the system are listed under **IPv4 network devices** and **IPv6 network devices** (only network interfaces in the `ethX` scheme are shown).

Network interfaces can be configured for IPv4 and/or IPv6. IPv4 addresses have a 32-bit length and are generally written in four blocks in decimal form (e.g., `192.0.2.10`), whereas IPv6 addresses are four times as long and typically written in hexadecimal form (e.g., `2001:0DB8:FE29:DE27:0000:0000:0000:0000`).

#### 8.2.4.1.1. Configuration of IPv4 addresses

Feedback

If the **Dynamic (DHCP)** option was not chosen, the IP address to be bound to the network card must be entered. In addition to the **IPv4 address** the **net mask** must also be entered. **DHCP query** is used to request

an address from a DHCP server. Unless the **Dynamic (DHCP)** option is activated, the values received from the DHCP request are configured statically.

Server systems can also be configured via DHCP. This is necessary for some cloud providers, for example. If the assignment of an IP address for a server fails, a random link local address (`169.254.x.y`) is configured as a replacement.

For UCS server systems the address received via DHCP is also written to the LDAP directory.

### Note

Not all services (e.g., DNS servers) are suitable for use on a DHCP-based server.

(UCR variables: `interfaces/ethX/address`, `interfaces/ethX/netmask`, `interfaces/ethX/type`)

Besides the physical interfaces, additional virtual interfaces can also be defined in the form `interfaces/ethX_Y/setting`.

#### 8.2.4.1.2. Configuration of IPv6 addresses

Feedback

The IPv6 address can be configured in two ways: Stateless address autoconfiguration (SLAAC) is employed in the **Autoconfiguration (SLAAC)** configuration. In this, the IP address is assigned from the routers of the local network segment. Alternatively, the address can also be configured statically by entering the **IPv6 address** and **IPv6 prefix**. In contrast to DHCP, in SLAAC there is no assignment of additional data such as the DNS server to be used. There is an additional protocol for this (DHCPv6), which, however, is not employed in the dynamic assignment. One network card can be used for different IPv6 addresses. The **Identifier** is a unique name for individual addresses. The main address always uses the identifier *default*; functional identifiers such as *Interface mail server* can be assigned for all other addresses.

(UCR variables: `interfaces/ethX/ipv6/address`, `interfaces/ethX/ipv6/prefix`, `interfaces/eth0/ipv6/acceptRA` activates SLAAC).

Further network settings can be performed under **Global network settings**.

The IP addresses for the standard gateways in the subnetwork can be entered under **Gateway (IPv4)** and **Gateway (IPv6)**. It is not obligatory to enter a gateway for IPv6, but recommended. A gateway configured here has preference over router advertisements, which might otherwise be able to change the route.

(UCR variables: `gateway`, `ipv6/gateway`)

#### 8.2.4.1.3. Configuring the name servers

Feedback

There are two types of DNS servers:

- An **External DNS Server** is employed for the resolution of host names and addresses outside of the UCS domain, e.g., `univention.de`. This is typically a name server operated by the Internet provider.
- A **Domain DNS Server** is a local name server in the UCS domain. This name server usually administrates host names and IP addresses belonging to the UCS domain. If an address is not found in the local inventory, an external DNS server is automatically requested. The DNS data are saved in the LDAP directory service, i.e., all domain DNS servers deliver identical data.

A local DNS server is set up on the master domain controller, backup domain controller and slave domain controller system roles. Here, you can configure which server should be primarily used for the name resolution by entering the **Domain DNS Server**.

(UCR variables: `nameserver1` to `nameserver3`, `dns/forwarder1` to `dns/forwarder3`,

#### 8.2.4.1.4. Configuration of bridges/bonding/VLANs

Feedback

UCS supports advanced network configurations using bridging, bonding and virtual networks (VLAN):

- Bridging is often used with virtualization to connect multiple virtual machines running on a host through one shared physical network interface.
- Bondings allows failover redundancy for hosts with multiple physical network interfaces to the same network.
- VLANs can be used to separate network traffic logically while using only one (or more) physical network interface.

##### 8.2.4.1.4.1. Prerequisite when using UCS Virtual Machine Manager

Feedback

When the application *KVM virtualization server* is installed, the network configuration is changed: A bridge with the name `br0` is configured, and the network device `eth0` is added. Additional network cards are not adapted accordingly.

When updating from UCS 3.2, the configuration prevents using the advanced network settings. It can be disabled by setting the following UCR variables:

```
# for KVM:
ucr set uvmm/kvm/bridge/autostart=no
# After that re-enable the support in the UMC basic settings dialog
ucr unset umc/modules/setup/network/disabled/by
```

After that the server must be rebooted. Existing virtual machines must be re-configured for the new interfaces names, which is described in Section 16.5.4. Updating UVMM profiles for new virtual machines is recommended and described in Section 16.6.1.

##### 8.2.4.1.4.2. Bridging

Feedback

The most common application scenario for *bridging* is the shared use of a physical network card by one or more virtual machines. Instead of one network card for each virtual machine and the virtualization server itself, all systems are connected via a shared uplink. A bridge can be compared with a switch implemented in software which is used to connect the individual hosts together. The hardware network adapter used is called a *bridge port*.

In order to configure a bridge, *Bridge* must be selected as the **Interface type** under **Add**. The **Name of new bridge interface** can be selected at will. Then click on **Next**.

The physical network card intended to act as the uplink can be selected under **Bridge ports**. In the typical scenario of connecting virtual machines via just one network card, there is no risk of a network loop. If the bridge is used to connect two Ethernet networks, the spanning tree protocol (STP) is employed to avoid network loops [1]. The **Forwarding delay** setting configures the waiting time in seconds during which information is collected about the network topology when a connection is being made via STP. If the bridge is used for connecting virtual machines to one physical network card, STP should be disabled by setting the value to 0. Otherwise problems may occur when using DHCP, as the packets sent during the waiting time are not forwarded.

The **Additional bridge options** input field can be used to configure arbitrary bridge parameters. This is only necessary in exceptional cases; an overview of the possible settings can be found on the manual page bridge-utils-interfaces(5).

[1] The Linux kernel only implements STP, not the Rapid STP or Multiple STP versions.

Clicking on **Next** offers the possibility of optionally assigning the bridge an IP address. This interface can then also be used as a network interface for the virtualization host. The options are the same as described in Section 8.2.4.1.1 and Section 8.2.4.1.2.

#### 8.2.4.1.4.3. Bonding

Feedback

*Bonding* can be used to bundle two (or more) physical network cards in order to increase the performance or improve redundancy in failover scenarios.

In order to configure a bonding, *Bonding* must be selected as the **Interface type** under **Add**. The **Name of the bonding interface** can be selected at will. Then click on **Next**.

The network cards which form part of the bonding interface are selected under **Bond slaves**. The network cards which should be given preference in failover scenarios (see below) can be selected via **Bond primary**.

The **Mode** configures the distribution of the network cards within the bonding:

- **balance-rr (0)** distributes the packets equally over the available network interfaces within the bonding one after the other. This increases performance and improves redundancy. In order to use this mode, the network switches used must support *link aggregation*.

- When **active-backup (1)** is used, only one network card is active for each bonding interface (in the default setting this is the network interface configured in **Bond primary**). If the primary network card fails, this is detected by the Linux kernel, which switches to another card in the bonding. This version increases redundancy. It can be used with every network switch.

In addition, there are also a number of other bonding methods. These are generally only relevant for special cases and are described under [bonding].

The Media Independent Interface (MII) of the network cards is used to detect failed network adapters. The **MII link monitoring frequency** setting specifies the testing interval in milliseconds.

All other bonding parameters can be configured under **Additional bonding options**. This is only necessary in exceptional cases; an overview of the possible settings can be found under [bonding].

Clicking on **Next** allows to optionally assign the bonding interface an IP address. If one of the existing network cards which form part of the bonding interface has already been assigned an IP address, this configuration will be removed. The options are the same as described in Section 8.2.4.1.1 and Section 8.2.4.1.2.

#### 8.2.4.1.4.4. VLANs

Feedback

VLANs can be used to separate the network traffic in a physical network logically over one or more virtual subnetworks. Each of these virtual networks is an independent broadcast domain. This makes it e.g. possible to differentiate between a network for the employees and a guest network for visitors in a company network although they use the same physical cables. The individual end devices can be assigned to the VLANs via the configuration of the switches. The network switches must support 802.1q VLANs.

A distinction is made between two types of connections between network cards:

- A connection only transports packets from a specific VLAN. In this case, *untagged* data packets are transmitted.

  This is typically the case if only one individual end device is connected via this network connection.

- A connection transports packets from several VLANs. This is also referred to as a *trunk link*. In this case, each packet is assigned to a VLAN using a VLAN ID. During transmission between trunk links and specific

VLANs, the network switch takes over the task of filtering the packets by means of the VLAN IDs as well as adding and removing the VLAN IDs.

This type of connection is primarily used between switches/servers.

Some switches also allow the sending of packets with and without VLAN tags over a shared connection, but this is not described in more detail here.

When configuring a VLAN in Univention Management Console it is possible to configure for a computer which VLANs it wants to participate in. An example here would be an internal company web server, which should be available both to the employees and any users of the guest network.

In order to configure a VLAN, *Virtual LAN* must be selected as the **Interface type** under **Add**. The network interface for which the VLAN is configured is specified with **Parent interface**. The **VLAN ID** is the unique identifier of the VLAN. Valid values are from 1 to 4095. Then **Next** must be clicked.

Clicking on **Next** allows to optionally assign the VLAN interface an IP address. The options are the same as described in Section 8.2.4.1.1 and Section 8.2.4.1.2. When assigning an IP address, ensure that the address matches the assigned VLAN address range.

### 8.2.4.2. Configuring proxy access

Feedback

The majority of the command line tools which access web servers (e.g., `wget`, `elinks` or `curl`) check whether the environment variable `http_proxy` is set. If this is the case, the proxy server set in this variable is used automatically.

The Univention Configuration Registry variable `proxy/http` can also be used to activate the setting of this environment variable via an entry in `/etc/profile`.

The proxy URL must be specified for this, e.g., `http://192.0.2.100`. The proxy port can be specified in the proxy URL using a colon, e.g., `http://192.0.2.100:3128`. If the proxy requires authentication for the accessing user, this can be provided in the form `http://username:password@192.0.2.100`.

The environment variable is not adopted for sessions currently opened. A relogin is required for the change to be activated.

The Univention tools for software updates also support operation via a proxy and query the Univention Configuration Registry variable.

Individual domains can be excluded from use by the proxy by including them separated by commas in the Univention Configuration Registry variable `proxy/no_proxy`. Subdomains are taken into account; e.g. an exception for `software-univention.de` also applies for `updates.software-univention.de`.

## 8.2.5. Configuration of the monitor settings

Feedback

The configuration of the graphic resolutions and monitor parameters is performed via automatic detection of the graphics card and the monitor in the default setting. When this is done, the best available driver for the graphics card is selected automatically and the monitor resolution set to the highest value supported by the monitor.

The settings can be set with a Univention Configuration Registry policy. Manual configuration is also necessary if dual monitor operation is to be used. The following provides a selection of the important settings and the corresponding UCR variables in parentheses:

- **Graphics adapter driver** selects the responsible Xorg driver (`xorg/device/driver`).

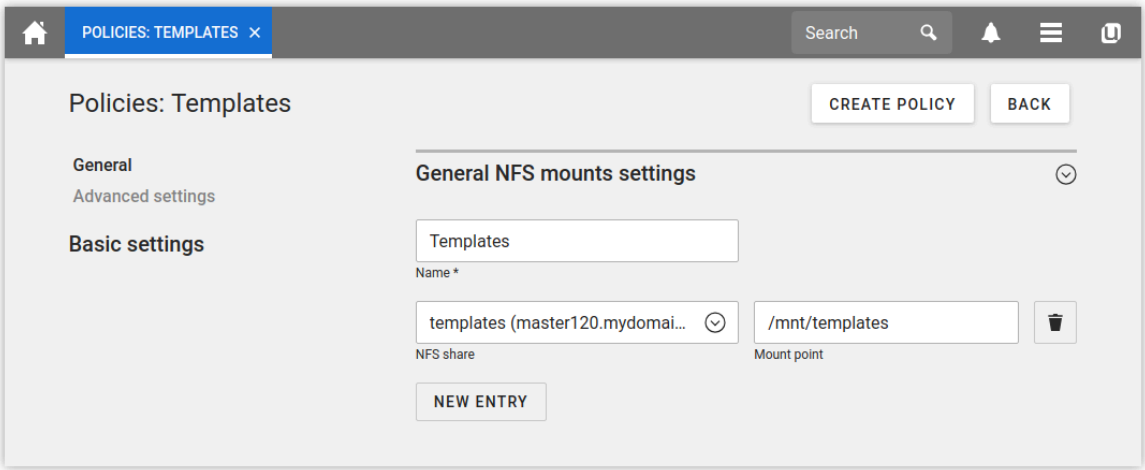
- The screen resolution of the main monitor should be entered under **Resolution of primary monitor**. The values for width and height in pixels should be separated by an 'x', e.g., `1024x768` (`xorg/resolution`).
- **Resolution of secondary display** defines the screen resolution of a second monitor, if present. This combines with the primary monitor to display a shared screen area (`xorg/resolution/secondary`).
- The **Position of secondary display** menu specifies the relative position of the secondary monitor with respect to the primary monitor (`xorg/display/relative-position`).
- The **Color depth** should be entered in bits per pixel. Admissible values are 1, 2, 4, 8, 16 and 24. (24-bit is true color color depth) (`xorg/screen/DefaultDepth`).

## 8.2.6. Mounting NFS shares

Feedback

The **NFS mounts** policy of the UMC computer management can be used to configure NFS shares, which are mounted on the system. There is a **NFS share** for selection, which is mounted in the file path specified under **Mount point**.

**Figure 8.5. Mounting a NFS share**




## 8.2.7. Collection of list of supported hardware

Feedback

Univention collects information about hardware which is compatible with UCS and in use by customers. The information processed for this is gathered by the UMC module **System information**.

All files are forwarded to Univention anonymously and only transferred once permission has been received from the user.

The start dialogue contains the entry fields **Manufacturer** and **Model**, which must be completed with the values determined from the DMI information of the hardware. The fields can also be adapted and an additional **Descriptive comment** added.

If the system information is transferred as part of a support request, the **This is related to a support case** option should be activated. A ticket number can be entered in the next field; this facilitates assignment and allows quicker processing.

Clicking on **Next** offers an overview of the transferred system information. In addition, a compressed .tar archive is created, which contains a list of the hardware components used in the system and can be downloaded via **Archive with system information**.

Clicking on **Next** again allows you to select the way the data are transferred to Univention. **Upload** transmits the data via HTTPS, **Send mail)** opens a dialogue, which lists the needed steps to send the archive via e-mail.

# 8.3. Administration of local system configuration with Univention Configuration Registry

Feedback

## 8.3.1. Introduction

Feedback

Univention Configuration Registry is the central tool for managing the local system configuration of a UCS-based system. Direct editing of the configuration files is usually not necessary.

Settings are specified in a consistent format in a registry mechanism, the so-called *Univention Configuration Registry variables*. These variables are used to generate the configuration files used effectively by the services/programs from the configuration templates (the so-called *Univention Configuration Registry templates*).

This procedure offers a range of advantages:

- It is not usually necessary to edit any configuration files manually. This avoids errors arising from invalid syntax of configuration settings or similar.
- There is a uniform interface for editing the settings and the different syntax formats of the configuration files are hidden from the administrator.
- Settings are decoupled from the actual configuration file, i.e., if a software uses a different configuration format in a new version, a new template in a new format is simply delivered instead of performing time-consuming and error-prone conversion of the file.
- The variables used in a configuration file administrated with Univention Configuration Registry are registered internally. This ensures that when a UCR variable is changed, all the configuration files containing the changed variable are recreated.

Univention Configuration Registry variables can be configured in the command line using the `univention-config-registry` command (short form: `ucr` or via Univention Management Console.

As the majority of packages perform their configuration via Univention Configuration Registry and the corresponding basic settings need to be set up during the installation, hundreds of Univention Configuration Registry variables are already set after the installation of a UCS system.

UCR variables can also be used efficiently in shell scripts for accessing current system settings.

The variables are named according to a tree structure with a forward slash being used to separate components of the name. For example, Univention Configuration Registry variables beginning with `ldap` are settings which apply to the local directory service.

A description is given for the majority of variables explaining their use.

If a configuration file is administrated by a UCR template and the required setting has not already been covered by an existing variable, the UCR template should be edited instead of the configuration file. If the configuration were directly adapted, the next time the file is regenerated - e.g., when a registered UCR variable is set - the local modification will be overwritten again. Adaptation of UCR templates is described in Section 8.3.5.

Part of the settings configured in Univention Configuration Registry are system-specific (e.g., the computer name); many settings can, however, be used on more then one computer. The Univention Configuration

Registry policy in the domain administration of Univention Management Console can be used to compile variables and apply them on more than one computer.

The evaluation of the Univention Configuration Registry variables on a UCS system comprises four stages:

- First the local Univention Configuration Registry variables are evaluated.
- The local variables are overruled by policy variables which are usually sourced from the directory service
- The `--schedule` option is used to set local variables which are only intended to apply for a certain period of time. This level of the Univention Configuration Registry is reserved for local settings which are automated by time-controlled mechanisms in Univention Corporate Server.
- When the `--force` option is used in setting a local variable, settings adopted from the directory service and variables from the schedule level are overruled and the given value for the local system fixed instead. An example:

```
univention-config-registry set --force mail/messagesizelimit=1000000
```

If a variable is set which is overwritten by a superordinate policy, a warning message is given.

The use of the Univention Configuration Registry policy is documented in the Section 8.3.4.

## 8.3.2. Using the Univention Management Console web interface

Feedback

The UMC module **Univention Configuration Registry** can be used to display and adjust the variables of a system. There is also the possibility of setting new variables using **Add new variable**.

A search mask is displayed on the start page. All variables are classified using a **Category**, for example all LDAP-specific settings.

The **Search attribute** can be entered as a filter in the search mask, which can refer to the variable name, value or description.

Following a successful search, the variables found are displayed in a table with the variable name and the value. A detailed description of the variable is displayed when moving the mouse cursor over the variable name.

Clicking on the icon with the stylized pen edits the setting of a variable. The icon with the stylized minus sign allows the deletion of a variable.

## 8.3.3. Using the command line front end

Feedback

The command line interface of Univention Configuration Registry is run using the `univention-config-registry` command. Alternatively, the short form `ucr` can be used.

### 8.3.3.1. Querying a UCR variable

Feedback

A single Univention Configuration Registry variable can be queried with the parameter `get`:

```
univention-config-registry get ldap/server/ip
```

The parameter `dump` can also be used to display all currently set variables:

```
univention-config-registry dump
```

#### 8.3.3.1.1. Setting UCR variables

Feedback

The parameter `set` is used to set a variable. The variable can be given any name consisting exclusively of letters, full stops, figures, hyphens and forward slashes.

```
univention-config-registry set VARIABLENAME=VALUE
```

If the variable already exists, the content is updated; otherwise, a new entry is created.

The syntax is not checked when a Univention Configuration Registry variable is set. The change to a variable results in all configuration files for which the variable is registered being rewritten immediately. The files in question are output on the console:

In doing so it must be noted that although the configuration of a service is updated, the service in question is not restarted automatically! The restart must be performed manually.

It is also possible to perform simultaneous changes to several variables in one command line. If these refer to the same configuration file, the file is only rewritten once.

```
univention-config-registry set \
    dns/forwarder1=192.0.2.2 \
    sshd/xforwarding="no" \
    sshd/port=2222
```

A conditional setting is also possible. For example, if a value should only be saved in a Univention Configuration Registry variable when the variable does not yet exist, this can be done by entering a question mark instead of the equals sign when assigning values.

```
univention-config-registry set dns/forwarder1?192.0.2.2
```

#### 8.3.3.1.2. Searching for variables and set values

Feedback

The *`search`* parameter can be used to search for a variable. This command searches for variable names which contain *nscd* and displays these with their current assignments:

```
univention-config-registry search nscd
```

Alternatively, searches can also be performed for set variable values. This request searches for all variables set to `master.example.com`:

```
univention-config-registry search --value master.example.com
```

Search templates in the form of regular expressions can also be used in the search. The complete format is documented at https://docs.python.org/2/library/re.html.

#### 8.3.3.1.3. Deleting UCR variables

Feedback

The parameter *`unset`* is used to delete a variable. The following example deletes the variable `dns/forwarder2`. It is also possible here to specify several variables to be deleted:

```
univention-config-registry unset dns/forwarder2
```

#### 8.3.3.1.4. Regeneration of configuration files from their template

Feedback

The *`commit`* parameter is used to regenerate a configuration file from its template. The name of the configuration file is entered as a parameter, e.g.:

```
univention-config-registry commit /etc/samba/smb.conf
```

As UCR templates are generally regenerated automatically when UCR variables are edited, this is primarily used for tests.

If no file name is given when running `ucr commit`, all of the files managed by Univention Configuration Registry will be regenerated from the templates. It is, however, not generally necessary to regenerate all the configuration files.

#### 8.3.3.1.5. Sourcing variables in shell scripts

Feedback

The parameter `shell` is used to display Univention Configuration Registry variables and their current assignments in a format that can be used in shell scripts.

```
univention-config-registry shell ldap/server/name
```

Different conversions are involved in this: forward slashes in variable names are replaced with underscores and characters in the values which have a particular significance in shell scripts are included in quotation marks to ensure they are not altered.

The Univention Configuration Registry output must be executed via the command `eval` for Univention Configuration Registry variables to be able to be read in a shell script as environment variables:

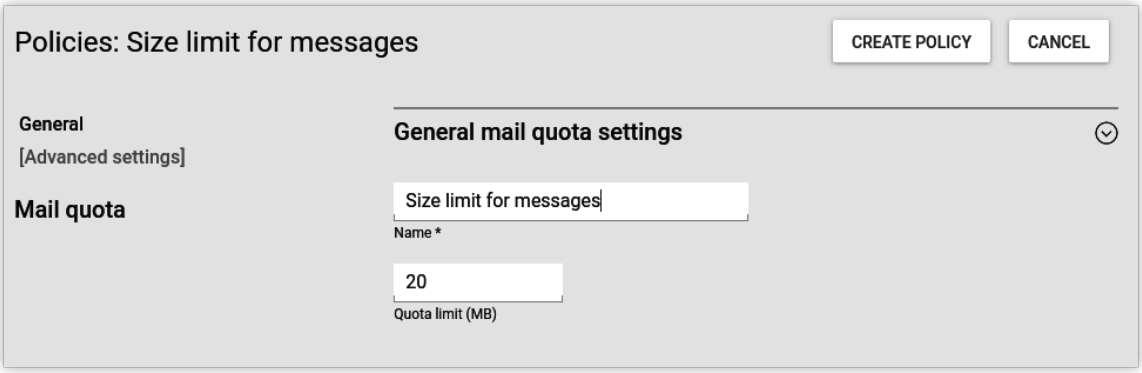
```
# eval "$(univention-config-registry shell ldap/server/name)"
# echo "$ldap_server_name"
master.firma.de
```

## 8.3.4. Policy-based configuration of UCR variables

Feedback

Part of the settings configured in Univention Configuration Registry are system-specific (e.g., the computer name); many settings can, however, be used on more then one computer. The **Univention Configuration Registry** policy managed in the UMC module **Policies** can be used to compile variables and apply them on more than one computer.

**Figure 8.6. Policy-based configuration of the maximum mail size**




Firstly, a **Name** must be set for the policy which is to be created, under which the variables will later be assigned to the individual computer objects.

In addition, at least one **Variable** must be configured and a **Value** assigned.

This policy can then be assigned to a computer object or a container/OU (see Section 4.6.2). Note that the evaluation of configured values differs from other policies: the values are not forwarded directly to the com-

puter, but rather written on the assigned computer by Univention Directory Policy. The time interval used for this is configured by the Univention Configuration Registry variable `ldap/policy/cron` and is set to hourly as standard.

## 8.3.5. Modifying UCR templates

Feedback

In the simplest case, a Univention Configuration Registry template is a copy of the original configuration file in which the points at which the value of a variable are to be used contain a reference to the variable name.

Inline Python code can also be integrated for more complicated scenarios, which then also allows more complicated constructions such as conditional assignments.

### Note

Univention Configuration Registry templates are included in the corresponding software packages as configuration files. When packages are updated, a check is performed for whether any changes have been made to the configuration files. If configuration files are no longer there in the form in which they were delivered, they will not be overwritten. Instead a new version will be created in the same directory with the ending `.debian.dpkg-new`. If changes are to be made on the Univention Configuration Registry templates, these templates are also not overwritten during the update and are instead re-saved in the same directory with the ending `.dpkg-new` or `.dpkg-dist`. Corresponding notes are written in the `/var/log/univention/actualise.log` log file. This only occurs if UCR templates have been locally modified.

The UCR templates are stored in the `/etc/univention/templates/files/` directory. The path to the templates is the absolute path to the configuration file with the prefixed path to the template directory. For example, the template for the `/etc/issue` configuration file can be found under `/etc/univention/templates/files/etc/issue`.

For the configuration files to be processed correctly by Univention Configuration Registry they must be in UNIX format. If configuration files are edited in DOS or Windows, for example, control characters are inserted to indicate line breaks, which can disrupt the way Univention Configuration Registry uses the file.

### 8.3.5.1. Referencing of UCR variables in templates

Feedback

In the simplest case, a UCR variable can be directly referenced in the template. The variable name framed by the string @%@ represents the wildcard. As an example the option for the activation of X11 forwarding in the configuration file `/etc/ssh/sshd_config` of the OpenSSH server:

```
X11Forwarding @%@sshd/xforwarding@%@
```

Newly added references to UCR variables are automatically evaluated by templates; additional registration is only required with the use of inline Python code (see Section 8.3.5.2).

### 8.3.5.2. Integration of inline Python code in templates

Feedback

Any type of Python code can be embedded in UCR templates by entering a code block framed by the string @!@. For example, these blocks can be used to realize conditional requests so that when a parameter is changed via a variable, further dependent settings are automatically adopted in the configuration file. The following code sequence configures for example network settings using the Univention Configuration Registry settings:

```
@!@
if configRegistry.get('apache2/ssl/certificate'):
    print 'SSLCertificateFile %s' % \
```

```
        configRegistry['apache2/ssl/certificate']
@!@
```

All the data output with the print function are written in the generated configuration file. The data saved in Univention Configuration Registry can be requested via the `ConfigRegistry` object, e.g.:

```
@!@
if configRegistry.get('version/version') and \
        configRegistry.get('version/patchlevel'):
    print 'UCS %(version/version)s-%(version/patchlevel)s' % \
        configRegistry
@!@
```

In contrast to directly referenced UCR variables (see Section 8.3.5.1), variables accessed in inline Python code must be explicitly registered.

The Univention Configuration Registry variables used in the configuration files are registered in *info* files in the `/etc/univention/templates/info/` directory which are usually named after the package name with the file ending `.info`. If new Python code is entered into the templates or the existing code changed in such a way that it requires additional or different variables, one of the existing `.info` files will need to be modified or a new one added.

Following the changing of `.info` files, the `ucr update` command must be run.

# 8.4. Basic system services

Feedback

This chapter describes basic system services of a UCS Installation such as the configuration of the PAM authentication framework, system logs and the NSCD.

## 8.4.1. Administrative access with the root account

Feedback

There is a `root` account on every UCS system for complete administrative access. The password is set during installation of the system. The root user is not stored in the LDAP directory, but instead in the local user accounts.

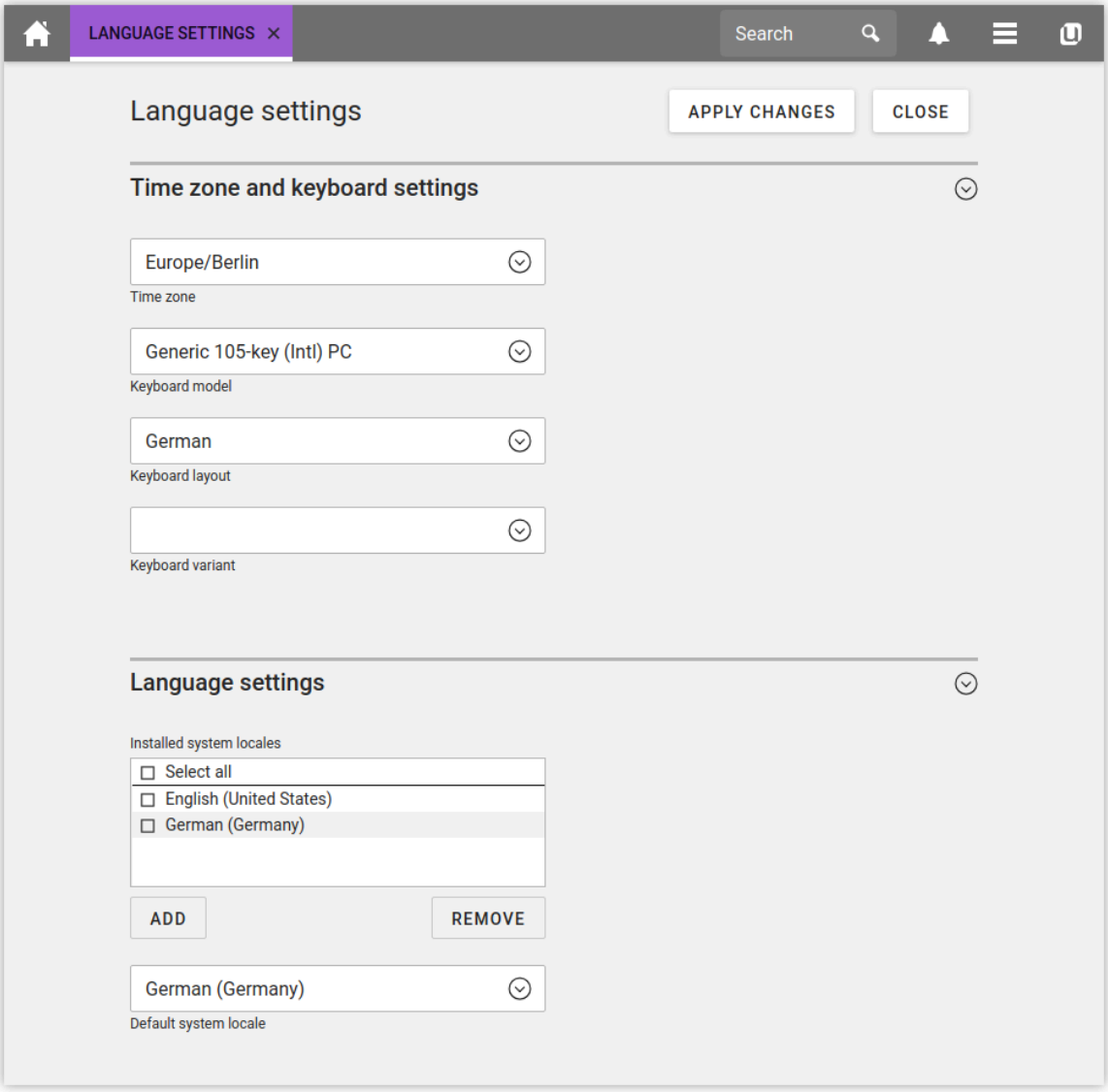
The password for the root user can be changed via the command line by using the `passwd` command. It must be pointed out that this process does not include any checks regarding either the length of the password or the passwords used in the past.

## 8.4.2. Configuration of language and keyboard settings

Feedback

In Linux, localization properties for software are defined in so-called *locales*. Configuration includes, among other things, settings for date and currency format, the set of characters in use and the language used for internationalized programs. The installed locales can be changed in Univention Management Console under **Language settings -> Installed system locales** The standard locale is set under **Default system locale**.

**Figure 8.7. Configuring the language settings**



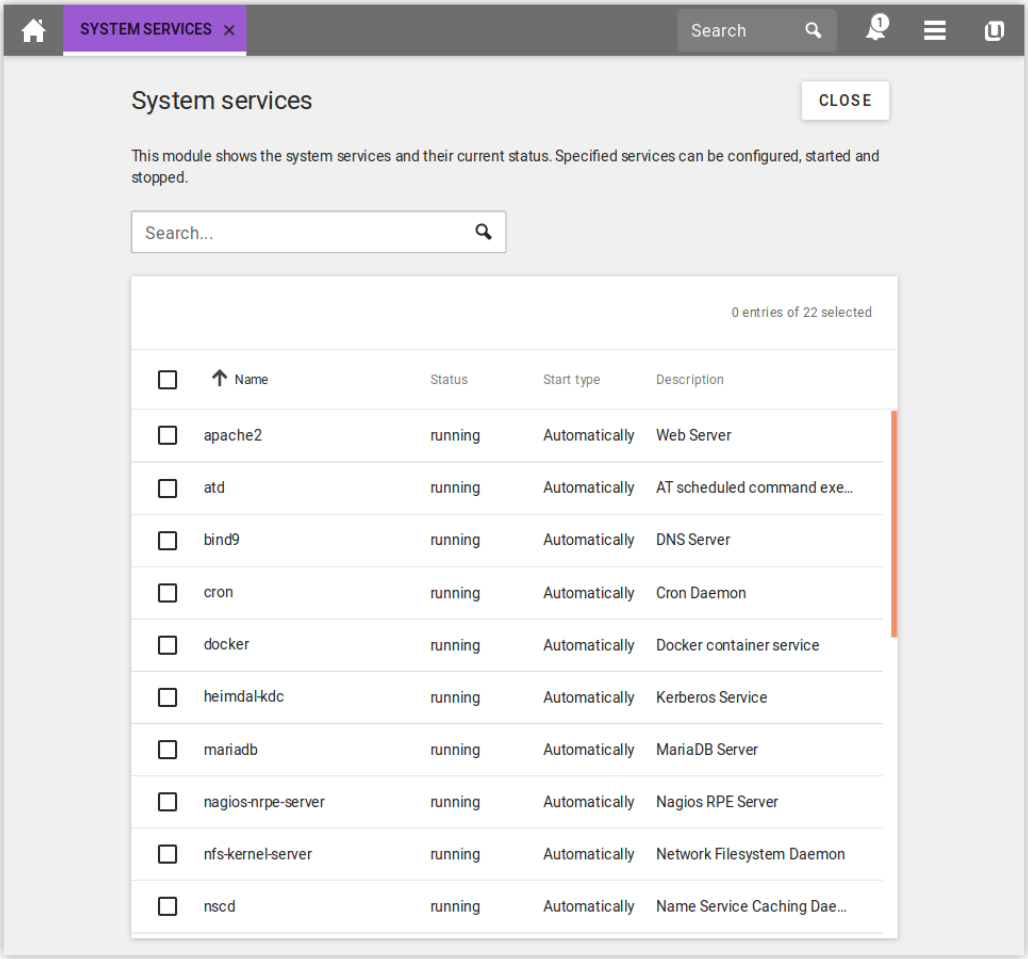

The **Keyboard layout** in the menu entry **Time zone and keyboard settings** is applied during local logins to the system.

## 8.4.3. Starting/stopping system services / configuration of automatic startup

Feedback

The UMC module **System services** can be used to check the current status of a system service and to start or stop it as required.

**Figure 8.8. Overview of system services**




In this list of all the services installed on the system, the current running runtime status and a **Description** are displayed under **Status**. The service can be started, stopped or restarted under **more**.

In the default setting, every service is started automatically when the system is started. In some situations, it can be useful not to have the service start directly, but instead only after further configuration. The action **Start manually** is used so that the service is not started automatically when the system is started, but can still be started subsequently. The action **Start never** also prevents subsequent service starts.

## 8.4.4. Authentication / PAM

Feedback

Authentication services in Univention Corporate Server are realized via *Pluggable Authentication Modules* (PAM). To this end different log-in procedures are displayed on a common interface so that a new log-in method does not require adaptation for existing applications.

### 8.4.4.1. Limiting authentication to selected users

Feedback

In the default setting, only the `root` user and members of the `Domain Admins` group can login remotely via SSH and locally on a `tty`.

This restriction can be configured with the Univention Configuration Registry variable `auth/SERVICE/restrict`. Access to this service can be authorized by setting the variables `auth/SERVICE/user/USERNAME` and `auth/SERVICE/group/GROUPNAME` to `yes`.

Login restrictions are supported for SSH (sshd), FTP (ftp), the login manager KDM (kdm), login on a tty (login), rlogin (rlogin), PPP (ppp) and other services (other). An example for SSH:

```
auth/sshd/group/Administrators: yes
auth/sshd/group/Computers: yes
auth/sshd/group/DC Backup Hosts: yes
auth/sshd/group/DC Slave Hosts: yes
auth/sshd/group/Domain Admins: yes
auth/sshd/restrict: yes
```

## 8.4.5. Configuration of the LDAP server in use

Feedback

Several LDAP servers can be operated in a UCS domain. The primary one used is specified with the Univention Configuration Registry variable `ldap/server/name`, further servers can be specified via the Univention Configuration Registry variable `ldap/server/addition`.

Alternatively, the LDAP servers can also be specified via a **LDAP server** policy in the UMC computer management. The order of the servers determines the order of the computer's requests to the server if a LDAP server cannot be reached.

In the default setting, only `ldap/server/name` is set following the installation or the domain join. If there is more than one LDAP server available, it is advisable to assign at least two LDAP servers using the **LDAP server** policy in order to improve redundancy. In cases of an environment distributed over several locations, preference should be given to LDAP servers from the local network.

## 8.4.6. Configuration of the print server in use

Feedback

The print server to be used can be specified with the Univention Configuration Registry variable `cups/server`.

Alternatively, the server can also be specified via the **Print server** policy in the UMC computer management.

## 8.4.7. Logging/retrieval of system messages and system status

Feedback

### 8.4.7.1. Log files

Feedback

All UCS-specific log files (e.g., for the listener/notifier replication) are stored in the `/var/log/univention/` directory. Services log in their own standard log files: for example, Apache to the file `/var/log/apache2/error.log`.

The log files are managed by logrotate. It ensures that log files are named in series in intervals (can be configured in weeks using the Univention Configuration Registry variable `log/rotate/weeks`, with the default setting being 12) and older log files are then deleted. For example, the current log file for the Univention Directory Listener is found in the `listener.log` file; the one for the previous week in `listener.log.1`, etc.

Alternatively, log files can also be rotated only once they have reached a certain size. For example, if they are only to be rotated once they reach a size of 50 MB, the Univention Configuration Registry variable `logrotate/rotates` can be set to `size 50M`.

The Univention Configuration Registry variable `logrotate/compress` is used to configure whether the older log files are additionally zipped with `gzip`.

### 8.4.7.2. Logging the system status

Feedback

`univention-system-stats` can be used to document the current system status in the `/var/log/univention/system-stats.log` file. The following values are logged:

- The free disk space on the system partitions (`df -lhT`)
- The current process list (`ps auxf`)
- Two `top` lists of the current processes and system load (`top -b -n2`)
- The current free system memory (`free`)
- The time elapsed since the system was started (`uptime`)
- Temperature, fan and voltage indexes from lm-sensors (`sensors`)
- A list of the current Samba connections (`smbstatus`)

The runtimes in which the system status should be logged can be defined in Cron syntax via the Univention Configuration Registry variable `system/stats/cron`, e.g., `0,30 * * * *` for logging every half and full hour. The logging is activated by setting the Univention Configuration Registry variable `system/stats` to `yes`. This is the default since UCS 3.0.

### 8.4.7.3. Querying system statistics in Univention Management Console

Feedback

The UMC module **Statistics** displays the utilization of system resources. For this purpose, a graph is displayed for different periods:

- The past 24 hours
- The past week
- The past month
- The past year

The following system information is documented:

- The utilization of the main memory in percent
- The processor utilization of the system
- The number of terminal server sessions active
- The utilization of the swap file

### 8.4.7.4. Process overview in Univention Management Console

Feedback

The UMC module **Process overview** displays a table of the current processes on the system. The processes can be sorted based on the following properties by clicking on the corresponding table header:

- CPU utilization in percent
- The user name under which the process is running
- Memory consumption in percent
- The process ID

The menu item **more** can be used to terminate processes. Two different types of termination are possible:

- The action **Terminate** sends the process a `SIGTERM` signal; this is the standard method for the controlled termination of programs.
- Sometimes, it may be the case that a program - e.g., after crashing - can no longer be terminated with this procedure. In this case, the action **Force terminate** can be used to send the signal `SIGKILL` and force the process to terminate.

As a general rule, terminating the program with `SIGTERM` is preferable as many programs then stop the program in a controlled manner and, for example, save open files.

#### 8.4.7.5. System error diagnosis in Univention Management Console

Feedback

The **System diagnostic** UMC module offers a corresponding user interface to analyze a UCS system for a range of known problems.

The module evaluates a range of problem scenarios known to it and suggests solutions if it is able to resolve the identified solutions automatically. This function is displayed via ancillary buttons. In addition, links are shown to further articles and corresponding UMC modules.

### 8.4.8. Executing recurring actions with Cron

Feedback

Regularly recurring actions (e.g., the processing of log files) can be started at a defined time with the Cron service. Such an action is known as a cron job.

#### 8.4.8.1. Hourly/daily/weekly/monthly execution of scripts

Feedback

Four directories are predefined on every UCS system, `/etc/cron.hourly/`, `/etc/cron.daily/`, `/etc/cron.weekly/` and `/etc/cron.monthly/`. Shell scripts which are placed in these directories and marked as executable are run automatically every hour, day, week or month.

#### 8.4.8.2. Defining local cron jobs in `/etc/cron.d/`

Feedback

A cron job is defined in a line, which is composed of a total of seven columns:

- Minute (0-59)
- Hour (0-23)
- Day (1-31)
- Month (1-12)
- Weekday (0-7) (0 and 7 both stand for Sunday)
- Name of user executing the job (e.g., root)
- The command to be run

The time specifications can be set in different ways. One can specify a specific minute/hour/etc. or run an action every minute/hour/etc. with an *. Intervals can also be defined, for example */2 as a minute specification runs an action every two minutes.

Some examples:

```
30 * * * * root /usr/sbin/jitter 600 /usr/share/univention-samba/slave-
sync
*/5 * * * * www-data /usr/bin/php -q /usr/share/horde/reminders.php
```

### 8.4.8.3. Defining cron jobs in Univention Configuration Registry

Feedback

Cron jobs can also be defined in Univention Configuration Registry. This is particularly useful if they are set via a Univention Directory Manager policy and are thus used on more than one computer.

Each cron job is composed of at least two Univention Configuration Registry variables. *JOBNAME* is a general description.

- `cron/JOBNAME/command` specifies the command to be run (required)
- `cron/JOBNAME/time` specifies the execution time (see Section 8.4.8.2) (required)
- As standard, the cron job is run as a user `root`. `cron/JOBNAME/user` can be used to specify a different user.
- If an e-mail address is specified under `cron/JOBNAME/mailto`, the output of the cron job is sent there per e-mail.
- `cron/JOBNAME/description` can be used to provide a description.

## 8.4.9. Name service cache daemon

Feedback

Data of the NSS service is cached by the *Name Server Cache Daemon* (NSCD) in order to speed up frequently recurring requests for unchanged data. Thus, if a repeat request occurs, instead of a complete LDAP request to be processed, the data are simply drawn directly from the cache.

Since UCS 3.1, the groups are no longer cached via the NSCD for performance and stability reasons; instead they are now cached by a local group cache, see Section 7.3.

The central configuration file of the (`/etc/nscd.conf`) is managed by Univention Configuration Registry.

The access to the cache is handled via a hash table. The size of the hash table can be specified in Univention Configuration Registry, and should be higher than the number of simultaneously used users/hosts. For technical reasons, a prime number should be used for the size of the table. The following table shows the standard values of the variables:

**Table 8.12. Default size of the hash table**

| Variable | Default size of the hash table |
|---|---|
| `nscd/hosts/size` | 6007 |
| `nscd/passwd/size` | 6007 |

With very big caches it may be necessary to increase the size of the cache database in the system memory. This can be configured through the Univention Configuration Registry variables `nscd/hosts/maxdbsize`, `nscd/group/maxdbsize` and `nscd/passwd/maxdbsize`.

As standard, five threads are started by NSCD. In environments with many accesses it may prove necessary to increase the number via the Univention Configuration Registry variable `nscd/threads`.

In the basic setting, a resolved group or host name is kept in cache for one hour, a user name for ten minutes. With the Univention Configuration Registry variables `nscd/group/positive_time_to_live` and `nscd/passwd/positive_time_to_live` these periods can be extended or diminished (in seconds).

From time to time it might be necessary to manually invalidate the cache of the NSCD. This can be done individually for each cache table with the following commands:

```
nscd -i passwd
nscd -i hosts
```

The verbosity of the log messages can be configured through the Univention Configuration Registry variable `nscd/debug/level`.

## 8.4.10. RDP login to systems using XRDP

Feedback

XRDP is a daemon running on Univention Corporate Server, which allows users to open a X session on the server using the Remote Desktop Protocol (RDP)[2]. This protocol is natively supported by Microsoft Windows and many other operating systems and thus does not require additional software to be installed on the client PCs.

### 8.4.10.1. Installation

Feedback

XRDP is available through the App Center (see Section 5.3) and can be installed using the corresponding Univention Management Console module App Center. It can be installed on multiple machines. After the installation the machine runs a XRDP server[3].

### 8.4.10.2. Configuration

Feedback

#### 8.4.10.2.1. Allowed users

Feedback

By default all users of the UNIX group `Domain Admins` are allowed to connect using RDP. The name of the group can be changed by setting the Univention Configuration Registry variable `xrdp/access/admins`.

#### 8.4.10.2.2. Maximum number of concurrent sessions

Feedback

The maximum number of concurrently running RDP sessions is by default limited to `10` and can be changed using the Univention Configuration Registry variable `xrdp/sessions/max`.

#### 8.4.10.2.3. Login window customization

Feedback

The title and the images used in the login windows can be configured using the Univention Configuration Registry variable `xrdp/title` and `xrdp/imagedir`. The directory must contain 4 images using the Windows bitmap format. The first two are used as the rectangular logo at the left side on the input fields, the other two are used for the banner displayed below the input window:

`ad24b.bmp`

24 bit color logo, 140 x 140

`ad256.bmp`

256 color logo, 140 x 140

`xrdp24b.bmp`

24 bit color banner, 256 x 192

`xrdp256.bmp`

256 color banner, 256 x 192

[2] https://en.wikipedia.org/wiki/Remote_Desktop_Protocol
[3] http://www.xrdp.org/

### 8.4.10.3. Client software

Feedback

- Microsoft Windows has a built-in RDP client. It will show a certificate warning as long as the UCS certificate has not been imported.
- UCS and many other Linux distributions include rdesktop[4] and FreeRDP[5].
- For Apple iOS there is iResktop[6].
- For Android there are multiple apps[7].

### 8.4.10.4. Known issue: Wrong keyboard layout

Feedback

When you log into KDE using XRDP, the keyboard is configured to use the US layout by default. To change this open the K-Menu, switch to the **Computer tab**, open **System settings**, then open the **Regional & Languages** in the Personal section. There open **Keyboard Layout**, select **Enable keyboard layouts** and add the requested layout from the left list of available layouts. Click the right arrow to add the layout and reorder it to the top in the right list of **Active layouts** using the up/down buttons. Apply the changes and close the application. For background information see the project documentation[8].

### 8.4.10.5. Alternatives

Feedback

VNC

Virtual Network Connect provides a simpler protocol, which is used by KVM for example.

SSH

Only provides a text console by default, but can be used to tunnel X sessions natively.

## 8.4.11. SSH login to systems

Feedback

When installing a UCS system, an SSH server is also installed per preselection. SSH is used for realizing encrypted connections to other hosts, wherein the identity of a host can be assured via a check sum. Essential aspects of the SSH server's configuration can be adjusted in Univention Configuration Registry.

By default the login of the privileged `root` user is permitted by SSH (e.g. for configuring a newly installed system where no users have been created yet, from a remote location).

- If the Univention Configuration Registry variable `sshd/permitroot` is set to `without-password`, then no interactive password request will be performed for the `root` user, but only a login based on a public key. By this means brute force attacks to passwords can be avoided.
- To prohibit SSH login completely, this can be deactivated by setting the Univention Configuration Registry variable `auth/sshd/user/root` to `no`.

The Univention Configuration Registry variable `sshd/xforwarding` can be used to configure whether an X11 output should be passed on via SSH. This is necessary, for example, for allowing a user to start a program with graphic output on a remote computer by logging in with `ssh -X TARGETHOST`. Valid settings are `yes` and `no`.

The standard port for SSH connections is port 22 via TCP. If a different port is to be used, this can be arranged via the Univention Configuration Registry variable `sshd/port`.

[4] http://www.rdesktop.org/
[5] http://www.freerdp.com/
[6] http://www.irdesktop.com/
[7] https://play.google.com/store/search?q=rdp+client%26c=apps
[8] https://github.com/FreeRDP/FreeRDP/wiki/Keyboard

## 8.4.12. Configuring the time zone / time synchronization

Feedback

The time zone in which a system is located can be changed in Univention Management Console under **Language settings -> Time zone**.

Asynchronous system times between individual hosts of a domain can be the source of a large number of errors: the reliability of log files is impaired; Kerberos operation is disrupted; the correct evaluation of the validity periods of passwords can be disturbed; etc.

Usually the master domain controller functions as the time server of a domain. With the Univention Configuration Registry variables `timeserver`, `timeserver2` and `timeserver3` external NTP servers can be included as time sources.

Manual time synchronization can be started by the command `ntpdate`.

Windows clients joined in a Samba 4 domain only accept signed NTP time requests. If the Univention Configuration Registry variable `ntp/signed` is set to `yes`, the NTP replies are signed by Samba 4.

# Chapter 9. Services for Windows

9.1. Introduction ..... 167
9.2. Operation of a Samba domain based on Active Directory ..... 168
9.2.1. Installation ..... 168
9.2.2. Services of a Samba domain ..... 168
9.2.2.1. Authentication services ..... 168
9.2.2.2. File services ..... 169
9.2.2.3. Print services ..... 169
9.2.2.4. Univention S4 connector ..... 169
9.2.2.5. Replication of directory data ..... 170
9.2.2.6. Synchronization of the SYSVOL share ..... 170
9.2.3. Configuration and management of Windows desktops ..... 170
9.2.3.1. Group policies ..... 170
9.2.3.2. Logon scripts / NETLOGON share ..... 176
9.2.3.3. Configuration of the file server for the home directory ..... 176
9.2.3.4. Roaming profiles ..... 176
9.3. Active Directory Connection ..... 177
9.3.1. Introduction ..... 177
9.3.2. UCS as a member of an Active Directory domain ..... 177
9.3.3. Setup of the UCS AD connector ..... 179
9.3.3.1. Basic configuration of the UCS AD Connector ..... 180
9.3.3.2. Importing the SSL certificate of the Active Directory ..... 182
9.3.3.3. Starting/Stopping the Active Directory Connection ..... 184
9.3.3.4. Functional test of basic settings ..... 184
9.3.3.5. Changing the AD access password ..... 184
9.3.4. Additional tools / Debugging connector problems ..... 185
9.3.4.1. `univention-adsearch` ..... 185
9.3.4.2. `univention-connector-list-rejected` ..... 185
9.3.4.3. Logfiles ..... 185
9.3.5. Details on preconfigured synchronization ..... 185
9.3.5.1. Containers and organizational units ..... 185
9.3.5.2. Groups ..... 186
9.3.5.3. Users ..... 187
9.4. Migrating an Active Directory domain to UCS using Univention AD Takeover ..... 187
9.4.1. Introduction ..... 187
9.4.2. Preparation ..... 188
9.4.3. Domain migration ..... 188
9.4.4. Final steps of the takeover ..... 191
9.4.5. Tests ..... 191
9.5. Trust relationships ..... 191

## 9.1. Introduction

Feedback

UCS can offer Active Directory (AD) services, be a member of an Active Directory domain or synchronize objects between Active Directory domains and a UCS domain.

For the purposes of Windows systems, UCS can assume the tasks of Windows server systems:

- Domain controller function / authentication services
- File services
- Print services

In UCS all these services are provided by Samba.

UCS supports the mostly automatic migration of an existing Active Directory domain to UCS. All users, groups, computer objects and group policies are migrated without the need to rejoin the Windows clients. This is documented in Section 9.4.

Microsoft Active Directory domain controllers cannot join the Samba domain. This functionality is planned at a later point in time.

Samba can not join an Active Directory Forest yet at this point.

Incoming trust relationships with other Active Directory domains are configurable. In this setup the external Active Directory domain trusts authentication decisions of the UCS domain (Windows trusts UCS) so that UCS users can log on to systems and Active Directory backed services in the Windows domain (see Section 9.5). Outgoing trusts with Active Directory domain (UCS trusts Windows) are not supported currently.

### Note

The usage of UCS as a Windows NT-compatible domain controller is unsupported since UCS 4.3.

# 9.2. Operation of a Samba domain based on Active Directory

Feedback

## 9.2.1. Installation

Feedback

Samba as an AD domain controller can be installed on all UCS domain controllers from the Univention App Center with the application *Active Directory-compatible domain controller*. Alternatively, the software package ***univention-samba4*** can be installed. On the system roles master domain controller and backup domain controller the ***univention-s4-connector*** package must also be installed (`univention-run-join-scripts` command must be run after installation). Additional information can be found in Section 5.6.

A Samba member server can be installed on UCS member servers from the Univention App Center with the application Windows-compatible Fileserver. Alternatively, the software package ***univention-samba*** can be installed (`univention-run-join-scripts` command must be run after installation). Additional information can be found in Section 5.6.

Samba supports the operation as a *read-only domain controller*. The setup is documented in [ext-doc-win].

## 9.2.2. Services of a Samba domain

Feedback

### 9.2.2.1. Authentication services

Feedback

User logins can only be performed on Microsoft Windows systems joined in the Samba domain. Domain joins are documented in Section 3.2.2.

Users who log on to a Windows system are supplied with a Kerberos ticket when they log on. The ticket is then used for the further authentication. This ticket allows access to the domain's resources.

Common sources of error in failed logins are:

- Synchronization of the system times between the Windows client and domain controller is essential for functioning Kerberos authentication. In the default setting, the system time is updated via NTP during system startup. This can also be done manually using the command `w32tm /resync`.

- DNS service records need to be resolved during login. For this reason, the Windows client should use the domain controller's IP address as its DNS name server.

### 9.2.2.2. File services

Feedback

A file server provides files over the network and allows concentrating the storage of user data on a central server.

The file services integrated in UCS support the provision of shares using the CIFS protocol (see Chapter 12). Insofar as the underlying file system supports Access Control Lists (ACLs) (can be used with `ext3`, `ext4` and `XFS`), the ACLs can also be used by Windows clients.

Samba Active Directory domain controllers can also provide file services. As a general rule, it is recommended to separate domain controllers and file/print services in Samba environments - the same as the Microsoft recommendations for Active Directory - that means using domain controllers for logins/authentication and member services for file/print services. This ensures that a high system load on a file server does not result in disruptions to the authentication service. For smaller environments in which it is not possible to run two servers, file and print services can also be run on a domain controller.

Samba supports the `CIFS` protocol and the successor `SMB2` to provide file services. Using a client which supports `SMB2` (as of `Windows Vista`, i.e., `Windows 7/8` too) improves the performance and scalability.

The protocol can be configured using the Univention Configuration Registry variable `samba/max/protocol`. It must be set on all Samba servers and then all Samba server(s) restarted.

- `NT1` configures `CIFS` (supported by all Windows versions)
- `SMB2` configures `SMB2` (supported as of `Windows Vista`/`Windows 7`)
- `SMB3` configures `SMB3` (supported as of `Windows 8`)

### 9.2.2.3. Print services

Feedback

Samba offers the possibility of sharing printers set up under Linux as network printers for Windows clients. The management of the printer shares and the provision of the printer drivers is described in Chapter 13.

Samba AD domain controllers can also provide print services. In this case, the restrictions described in Section 9.2.2.2 must be taken into consideration.

### 9.2.2.4. Univention S4 connector

Feedback

When using Samba as an Active Directory domain controller, Samba provides a separate LDAP directory service. The synchronization between the UCS LDAP and the Samba LDAP occurs via an internal system service, the Univention S4 connector. The connector is enabled on the master domain controller by default and typically requires no further configuration.

Further information on the status of the synchronization can be found in the log file `/var/log/univention/connector-s4.log`. Additional information on analyzing connector replication problems can be found in SDB 1235.

The `univention-s4search` command can be used to search in the Samba directory service. If it is run as the `root` user, the required credentials of the machine account are used automatically:

```
root@master:~# univention-s4search sAMAccountName=Administrator
```

http://sdb.univention.de/1235

```
# record 1
dn: CN=Administrator,CN=Users,DC=example,DC=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: Administrator
instanceType: 4
(..)
```

#### 9.2.2.5. Replication of directory data

Feedback

Samba AD domains use the Directory Replication System (DRS) to replicate the directory data. DRS allows multi-master replication, i.e., the write changes from multiple domain controllers are synchronized at protocol level. Consequently, the use of snapshots in virtualization solutions should be avoided when using Samba 4 and Samba 4 should be operated on a server which is never switched off.

The complexity of the multi-master replication increases with each additional Samba AD domain controller. Consequently, it must be checked whether additional Samba AD domain controllers are necessary or if a member server would not be a better choice for new servers.

Additional information on troubleshooting replication problems can be found in SDB 1235.

#### 9.2.2.6. Synchronization of the SYSVOL share

Feedback

The SYSVOL share is a share which provides group policies and logon scripts in Active Directory / Samba 4. It is synchronized among all domain controllers and stored in the `/var/lib/samba/sysvol/` directory.

In Microsoft Active Directory, the SYSVOL share is synchronized by the File Replication Service (introduced with Windows 2000) or the Distributed File System (as of `Windows 2008 R2`). These replication methods are not yet fully implemented in Samba 4. The synchronization between the Samba 4 domain controllers is performed in UCS via a Cron job (every five minutes as standard - can be configured using the Univention Configuration Registry variable `samba4/sysvol/sync/cron`).

### 9.2.3. Configuration and management of Windows desktops

Feedback

#### 9.2.3.1. Group policies

Feedback

##### 9.2.3.1.1. Introduction

Feedback

Group policies are an Active Directory feature which allows the central configuration of settings for computers and users. Group policies are also supported by Samba AD domains. The policies only apply to Windows clients; Linux or Mac OS systems cannot evaluate the policies.

Group policies are often referred to as GPOs (*group policy objects*). Put more precisely, a GPO can contain a series of policies. Despite their name, group policy objects cannot be assigned directly to certain user groups, but instead are linked with certain AD administration units (domains, sites or organizational units) in the Samba directory service (Samba DS/AD) and thus refer to subordinate objects. A group-specific or user-specific evaluation is only indirectly possible via the *Security Filtering* of a group policy object, in which the *Apply group policy Allow/Deny* privilege can be directly restricted to certain groups, users or computers.

As a basic rule, a distinction must be made between *group policies* (GPOs) and the similarly named *group policy preferences (GPPs)*:

http://sdb.univention.de/1235

- The settings made via *GPOs* are binding, whereas *GPPs* are merely used to enter preferences in the registry of Windows clients, which can still be overwritten on the client in certain circumstances.

- The settings made via *GPOs* are also dynamically applied to the target objects, whereas, in contrast, the settings made via *GPPs* are entered statically in the registry of Windows clients (this is also referred to as *tattooing*).

For these reasons, *GPOs* are preferable to *GPPs* in the majority of cases. This remainder of this section deals exclusively with *GPOs*.

In contrast to UCS policies (see Section 4.6), group policies are not configured in Univention Management Console, but instead are configured in a separate editor, the *Group Policy Management* editor, which is a component of the *Remote Server Administration Tools (RSAT)*. The installation is described in Section 9.2.3.1.2.

There are two types of policies:

- *User policies* configure a user's settings, e.g., the configuration of the desktop. It is also possible to configure applications via group policies (e.g., the start page of Microsoft Internet Explorer or settings in LibreOffice).

- *Computer policies* define a Windows client's settings.

Computer policies are evaluated for the first time the computer starts up; user policies during login. The policies are also continually evaluated for logged in users / running systems and updated (every 90-120 minutes in the default setting. The period varies at random to avoid peak loads.)

The command `gpupdate /force` can also be run specifically to start the evaluation of group policies.

Some policies - e.g., for the installation of software or for login scripts - are only evaluated during login (user policies) or system startup (computer policies).

The majority of group policies only set one value in the Windows registry, which is then evaluated by Windows or an application. As standard users cannot modify any settings in the corresponding section of the Windows registry, it is also possible to configure restricted user desktops in which, for example, users cannot open the Windows Task Manager.

The group policies are stored in the SYSVOL share, see Section 9.2.2.6. They are linked with user and host accounts in the Samba directory service.

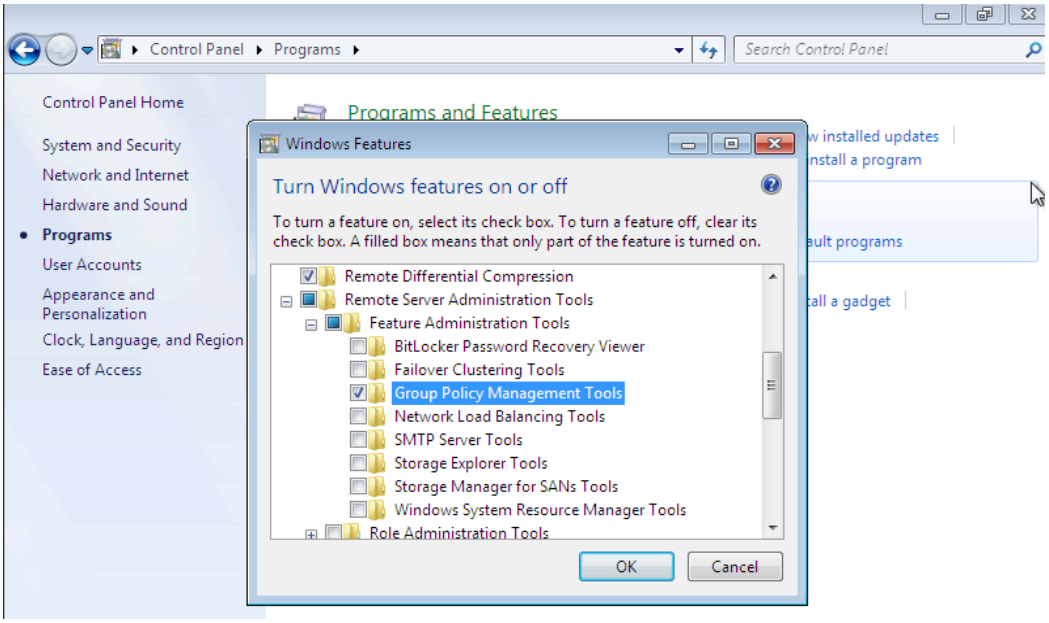
#### 9.2.3.1.2. Installation of Group Policy Management

Feedback

Group Policy Management can be installed as a component of the *Remote Server Administration Tools* on Windows clients. They can be found at [1] for Windows 7, at *Remote Server Administration Tools (RSAT) for Windows 8* [2] for Windows 8 or at *Remote Server Administration Tools (RSAT) for Windows 10* [3] for Windows 10.

[1] http://www.microsoft.com/en-us/download/details.aspx?id=7887
[2] http://www.microsoft.com/de-de/download/details.aspx?id=28972
[3] https://www.microsoft.com/en-us/download/details.aspx?id=45520

**Figure 9.1. Activating the Group Policy Management tools**




Following the installation, Group Policy Management must still be enabled in the Windows Control Panel. This is done by enabling the **Group Policy Management Tools** option under **Start -> Control Panel -> Programs -> Turn Windows features on or off -> Remote Server Administration Tools -> Feature Administration Tools**.

Following the enabling, Group Policy Management can be run under **Start -> Administrative Tools -> Group Policy Management** .

### 9.2.3.1.3. Configuration of policies with Group Policy Management

Feedback

Group policies can only be configured by users who are members of the `Domain Admins` group (e.g., the `Administrator`). When logging in, attention must be paid to logging in with the domain Administrator account and not the local Administrator account. Group Policy Management can be run on any system in the domain.

If more than one Samba domain controller is in use, consideration must be given to the replication of the GPO data, see Section 9.2.3.1.4.

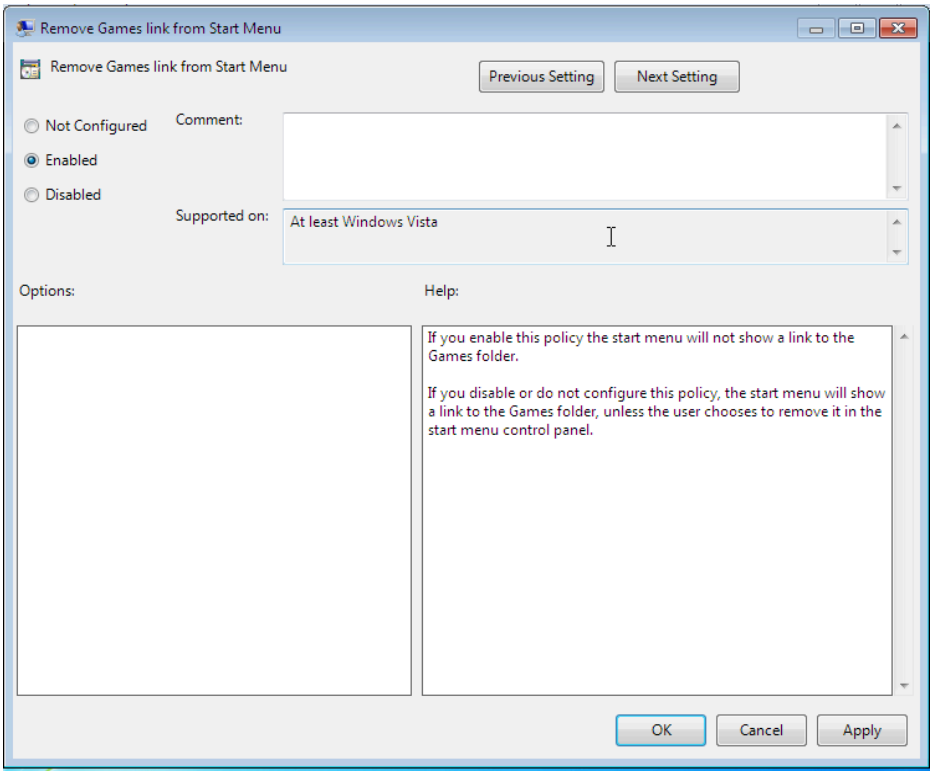
There are two basic possibilities for creating GPOs:

- They can be created in the **Group Policy Objects** folder and then linked to different positions in the LDAP. This is practical if a policy is to be linked to several positions in the LDAP.

- The GPO can also be created at an LDAP position ad hoc and then directly linked to it. This is the simpler means for small and medium-sized domains. Domains created ad hoc are also shown in the **Group Policy Objects** folder.

A policy can have one of three statuses: enabled, disabled or unset. The effect is always based on the formulation of the policy. For example, if it says **Disable feature xy**, the policy must be enabled to switch off the feature. Some policies have additional options, for example the **Enable mail quota** policy could include an additional option for managing the storage space.

**Figure 9.2. Editing a policy**




Two standard policy objects are predefined:

- The *Default Domain Policy* object can be used to configure global policies for all users and computers within the same domain.

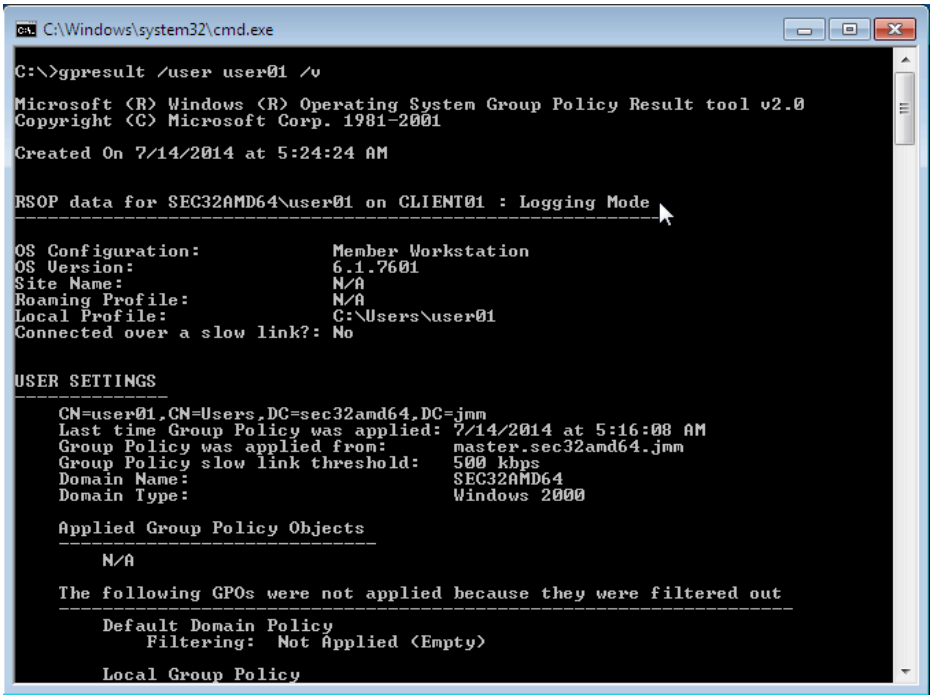
- The *Default Domain Controllers Policy* object has no use in a Samba domain (in a Microsoft AD domain the policies for Microsoft domain controllers would be performed via this object). The configuration of the Samba domain controllers in UCS is largely performed via Univention Configuration Registry.

AD domains can be structured in sites. All the sites are listed in the main menu of Global Policy Management. There is also a list of the domains there. The current Samba versions do not support forest domains, so there is only ever one domain displayed here.

One domain can be structured in different organizational units (OUs). This can, for example, be used to store the employees from accounting and the users in the administration department in different LDAP positions.

Group policies can mutually overlap. In this case, the inheritance principle applies, e.g., the superordinate policies overwrite the subordinate ones. The applicable policies for a user can be displayed on the Windows client either with the modeling wizard in Group Policy Management or by entering the command `gpresult /user USERNAME /v` in the Windows command line.

**Figure 9.3. Evaluating the GPO for the user `user01`**




The policies are evaluated in the following order:

- In the default setting, *Default Domain Policy* settings apply for all the users and computers within the domain.
- Policies linked to an OU overwrite policies from the default domain policy. If the OUs are nested further, in the case of conflict, the "most subordinate" policies in each case, in other words the one most closely linked to the target object, apply. The following evaluation order applies:
  - Assignment of a policy to an Active Directory site
  - Settings of the default domain policy
  - Assignment of a policy to an organizational unit (OU) (in turn, each subordinate OU overrules policies from superordinate OUs).

Example: A company blocks access to the Windows Task Manager in general. This is done by enabling the **Remove Task Manager** policy in the *Default Domain Policy* object. However, the Task Manager should still be available to some staff with the requisite technical expertise. These users are saved in the *IT staff* OU. An additional group policy object is now created in which the **Remove Task Manager** policy is set to *disabled*. The new GPO is linked with the *IT staff* OU.

#### 9.2.3.1.4. Configuration of group policies in environments with more than one Samba domain controller

Feedback

A group policy is technically composed of two parts: On the one hand there is a directory in the domain controllers' file system which contains the actual policy files which are to be implemented on the Window system (saved in the SYSVOL share (see Section 9.2.2.6)). On the other hand there is an object with the same name in the LDAP tree of the Samba directory service (Samba DS/AD), which is usually saved below an LDAP container named *Group Policy Objects*.

Although the LDAP replication between the domain controllers is performed in just a few seconds, the files in the SYSVOL share are only replicated every five minutes in the default setting. It must be noted that the application of newly configured group policies in this period may fail if a client happens to consult a domain controller which has not yet replicated the current files.

#### 9.2.3.1.5. Administrative templates (ADMX/ADM)

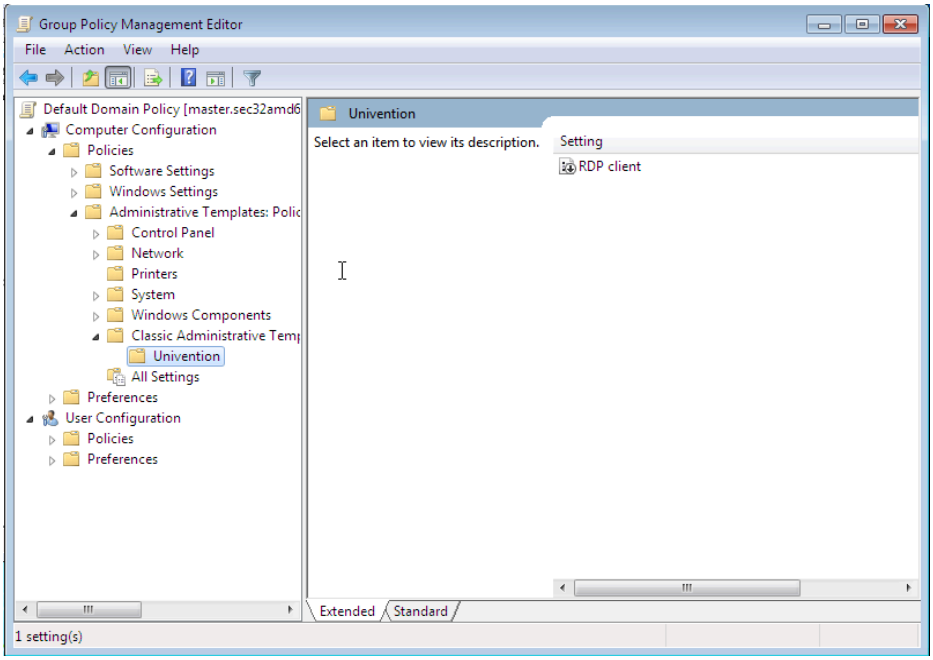


The policies displayed in Group Policy Management can be expanded with so-called *administrative templates*. This type of template defines the name under which the policy should appear in Group Policy Management and which value should be set in the Windows registry. Administrative templates are saved in so-called *ADMX files* (previously *ADM files*) [admx-reference]. Among other things, ADMX files offer the advantage that they can be provided centrally across several domain controllers so that Group Policy Management on all Windows clients displays the same configuration possibilities [admx-central].

The following example of an ADM file defines a computer policy in which a registry key is configured for the (fictitious) Univention RDP client. ADM files can also be converted to the newer ADMX format using third-party tools. Further information on the format of ADM files can be found under [microsoft-adm-templates] and [adm-templates-howto]. The administrative template must have the file suffix `.adm`:

```
CLASS MACHINE
CATEGORY "Univention"
POLICY "RDP client"
KEYNAME "Univention\RDP\StorageRedirect"
EXPLAIN "If this option it activated, sound output is enabled in the RDP
 client"
VALUENAME "Sound redirection"
VALUEON "Activated"
VALUEOFF "Deactivated"
END POLICY
END CATEGORY
```

**Figure 9.4. The activated administrative template**




The ADM file can then be converted to the ADMX format or imported directly via Group Policy Management. This is done by running the **Add/Remove Templates** option in the **Administrative templates** context menu.

**Add** can be used to import an ADM file. The administrative templates are also saved in the SYSVOL share and replicated, which allows Group Policy Management to access them from the Windows clients.

#### 9.2.3.1.6. Application of policies based on computer properties (WMI filters)

Feedback

It is also possible to configure policies based on system properties. These properties are provided via the Windows Management Instrumentation interface. The mechanism which builds on this is known as *WMI filtering*. This makes it possible, for example, to apply a policy only to PCS with a 64-bit processor architecture or with at least 8 GB of RAM. If a system property changes (e.g., if more memory is installed), the respective filter is automatically re-evaluated by the client.

The WMI filters are displayed in the domain structure in the **WMI Filters** container. **New** can be used to define an additional filter. The filter rules are defined under **Queries**. The rules are defined in a syntax similar to SQL. Examples rules can be found in [microsoft-wmi-filter] and [add-wmi-filters].

### 9.2.3.2. Logon scripts / NETLOGON share

Feedback

The NETLOGON share serves the purpose of providing logon scripts in Windows domains. The logon scripts are executed following after the user login and allow the adaptation of the user's working environment. Scripts have to be saved in a format which can be executed by Windows, such as `bat`.

The logon scripts are stored in `/var/lib/samba/sysvol/`*`Domainname`*`/scripts/` and provided under the share name *NETLOGON*. The file name of the script must be given relative to that directory.

The NETLOGON share is replicated within the scope of the SYSVOL replication.

The logon script can be assigned for each user, see Section 6.1.

### 9.2.3.3. Configuration of the file server for the home directory

Feedback

The home directory can be defined user-specifically in Univention Management Console, see Section 6.1. This is performed with the setting **Windows home path**, e.g., `\\ucs-file-server\smith`.

The multi edit mode of Univention Management Console can be used to assign the home directory to multiple users at one time, see Section 4.4.3.3.

### 9.2.3.4. Roaming profiles

Feedback

Samba supports roaming profiles, i.e., user settings are saved on a central server. This directory is also used for storing the files which the user saves in the *My Documents* folder. Initially, these files are stored locally on the Windows computer and then synchronized onto the Samba server when the user logs off.

If the profile path is changed in Univention Management Console, then a new profile directory will be created. The data in the old profile directory will be kept. These data can be manually copied or moved to the new profile directory. Finally, the old profile directory can be deleted.

No roaming profiles are used in the default setting in Samba 4.

Roaming profiles can be configured via a group policy found under **Computer configuration -> Policies -> Administrative templates -> System -> User profiles -> Set roaming profile path for all users logging onto this computer**.

#### Note

As standard, the Administrator accesses shares with root rights. If as a result the profile directory is created with the root user, it should be manually assigned to the Administrator with the command `chown`.

# 9.3. Active Directory Connection

Feedback

## 9.3.1. Introduction

Feedback

Univention Corporate Server can be operated together with an existing Active Directory domain (AD domain) in two different ways. Both modes can be set up using the *Active Directory Connection* application from the Univention App Center (see Section 5.6). This is available on a master domain controller and backup domain controller.

The two modes are:

- UCS as a part (domain member) of an AD domain (see Section 9.3.2)
- Synchronization of account data between an AD domain and a UCS domain (see Section 9.3.3).

In both modes, the Active Directory Connection service is used in UCS (UCS AD Connector for short), which can synchronize the directory service objects between a Windows 2012/2016/2019 server with Active Directory (AD) and the OpenLDAP directory of Univention Corporate Server.

In the first case, the configuration of a UCS server system as a member of an AD domain, the AD functions as the primary directory service and the respective UCS system joins the trust context of the AD domain. The domain membership gives the UCS system restricted access to the account data of the Active Directory domain. The set-up of this operating mode is described in detail in Section 9.3.2.

The second mode, which can be configured via the *Active Directory Connection* app, is used to run the UCS domain parallel to an existing AD domain. In this mode, each domain user is assigned a user account with the same name in both the UCS and the AD domain. Thanks to the use of the name identity and the synchronization of the encrypted password data, this mode allows transparent access between the two domains. In this mode, the authentication of a user in the UCS domain occurs directly within the UCS domain and as such is not directly dependent on the AD domain. The set-up of this operating mode is described in detail in Section 9.3.3.

## 9.3.2. UCS as a member of an Active Directory domain

Feedback

In the configuration of a UCS server system as a member of an AD domain (*AD member* mode), the AD functions as the primary directory service and the respective UCS system joins the trust context of the AD domain. The UCS system is not able to operate as an Active Directory domain controller itself. The domain membership gives the UCS system restricted access to the account data of the Active Directory domain, which it exports from the AD by means of the UCS AD Connector and writes locally in its own OpenLDAP-based directory service. In this configuration, the UCS AD Connector does not write any changes in the AD.

The *AD member* mode is ideal for expanding an AD domain with applications that are available on the UCS platform. Apps installed on the UCS platform can then be used by the users of the AD domain. The authentication is still performed against native Microsoft AD domain controllers.

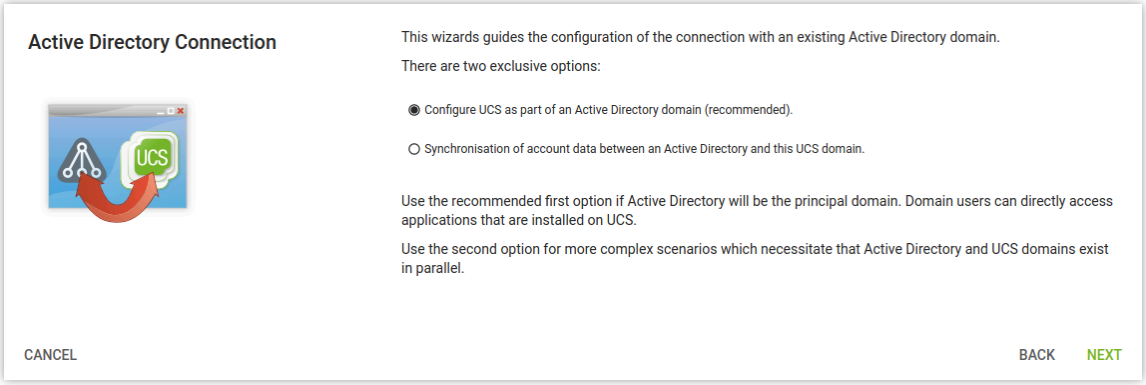
The set-up wizard can be started directly from the UCS installation by selecting *Join into an existing Active Directory domain*. Subsequently, the set-up wizard can be installed with the app *Active Directory Connection* from the Univention App Center. Alternatively, the software package ***univention-ad-connector*** can be installed. Further information can be found in Section 5.6.

### Note

- The *AD member* mode can only be configured on a master domain controller.
- The name of the DNS domain of the UCS systems must match that of the AD domain. The host name must of course be different.

- All the AD and UCS servers in a connector environment must use the same time zone.

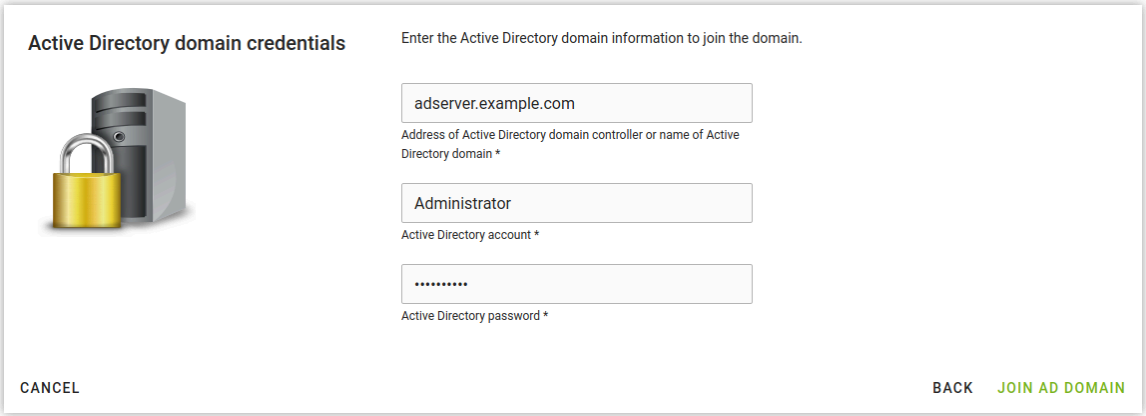
**Figure 9.5. Configuration of the operating mode as part of an AD domain**




In the first dialogue window of the set-up wizard, the point *Configure UCS as part of an AD domain* is preselected and can be confirmed with **[Next]**.

The next dialogue window requests the address of an AD domain controller as well as the name of the standard administrator account of the AD domain and its password. The standard AD administrator account should be used here. The specified AD domain controller should also provide DNS services for the domain. Pressing the **[Join AD domain]** button starts the domain join.

**Figure 9.6. Domain join of an AD domain**




If the system time of the UCS system is more than 5 minutes ahead of the system time of the AD domain controller, manual adjustment of the system times is required. This is necessary because the AD Kerberos infrastructure is used for the authentication. System times should not, however, be turned back, in order to avoid inconsistencies.

The domain join is performed automatically. The subsequent dialogue window should be confirmed with **[Finish]**. Then the UMC server should be restarted by clicking **[Restart]**.

## Note

Once the *AD member* mode has been set up, the authentication is performed against the AD domain controller. *Consequently, the password from the AD domain now applies for the administrator.* If an AD domain with a non-English language convention has been joined, the `administrator`

account from UCS is automatically changed to the spelling of the AD during the domain join. The same applies for all user and group objects with *Well Known SID* (e.g., `Domain Admins`).

### Warning

If additional UCS systems were already part of the UCS domain in addition to the master domain controller, they must also join the domain anew. At the same time they recognize that the master domain controller is in *AD member* mode and also join the authentication structure of the AD domain and can then also provide Samba file shares, for example.

### Note

As the AD Kerberos infrastructure is used for the authentication of users in this mode, it is essential that the system times of UCS and the AD domain controller are synchronized (with a tolerance of 5 minutes). For this purpose, the AD domain controller is configured as the NTP time server in UCS. In the case of authentication problems, the system time should always be the first thing to be checked.

Following this set-up, the *Active Directory Connection* UMC module can be used for further administration, e.g., for checking whether the service is running and to restart it if necessary (see Section 9.3.3.3).

To use an encrypted connection between Active Directory and the master domain controller not only for the authentication, but also for data exchange itself, the root certificate of the certification authority can be exported from the AD domain controller and uploaded via the UMC module. Further information on this topic is available in Section 9.3.3.2.

In the default setting, the Active Directory connection set up in this way does not transfer any password data from AD to the UCS directory service. Some apps from the Univention App Center require encrypted password data. If an app needs it, a note is shown in the App Center.

In *AD member* mode, in the default setting, the UCS AD Connector exports object data from the AD with the authorizations of the master domain controller's machine account. These authorizations are not sufficient for exporting encrypted password data. In this case, the LDAP DN of a privileged replication user can be adjusted manually in the Univention Configuration Registry variable `connector/ad/ldap/binddn`. This must be a member of the `Domain Admins` group in the AD. The corresponding password must be saved in a file on the master domain controller and the file name entered in the Univention Configuration Registry variable `connector/ad/ldap/bindpw`. If the access password is changed at a later point in time, the new password must be entered in this file. The access rights for the file should be restricted so that only the `root` owner has access.

The following commands demonstrate the steps in an example:

```
ucr set connector/ad/ldap/binddn=Administrator
ucr set connector/ad/ldap/bindpw=/etc/univention/connector/password
touch /etc/univention/connector/password
chmod 600 /etc/univention/connector/password
echo -n "Administrator password" > /etc/univention/connector/password
ucr set connector/ad/mapping/user/password/kinit=false
```

If desired, the AD domain controller can also be replaced by the master domain controller at a later point in time. This is possible via the *Active Directory Takeover* application (see Section 9.4).

## 9.3.3. Setup of the UCS AD connector

Feedback

As an alternative to membership in an AD domain, as described in the previous section, the Active Directory Connection can be used to synchronize user and group objects between a UCS domain and an AD domain. In

addition to unidirectional synchronization, this operating mode also allows bidirectional synchronization. In this operating mode, both domains exist in parallel and their authentication systems function independently. The prerequisite for this is the synchronization of the encrypted password data.

In the default setting, containers, organizational units, users, groups and computers are synchronized.

Information on the attributes configured in the basic setting and particularities to take into account can be found in Section 9.3.5.

The identical user settings in both domains allow users to access services in both environments transparently. After logging on to a UCS domain, subsequent connection to a file share or to an Exchange server with Active Directory is possible without a renewed password request. Users and administrators will find users and groups of the same name on the resources of the other domain and can thus work with their familiar permission structures.

The initialization is performed after the first start of the connector. All the entries are read out of the UCS, converted to AD objects according to the mapping set and added (or modified if already present) on the AD side. All the objects are then exported from the AD and converted to UCS objects and added/modified accordingly on the UCS side. As long as there are changes, the directory service servers continue to be requested. The UCS AD connector can also be operated in a unidirectional mode.

Following the initial sync, additional changes are requested at a set interval. This value is set to five seconds and can be adjusted manually using the Univention Configuration Registry variable `connector/ad/poll/sleep`.

If an object cannot be synchronized, it is firstly reset ("rejected"). Following a configurable number of cycles – the interval can be adjusted using the Univention Configuration Registry variable `connector/ad/retryrejected` – another attempt is made to import the changes. The standard value is ten cycles. In addition, when the UCS AD Connector is restarted, an attempt is also made to synchronize the previously rejected changes again.

The UCS AD connector can only be installed on a master domain controller or backup domain controller system.

#### 9.3.3.1. Basic configuration of the UCS AD Connector

Feedback

The UCS AD Connector is configured using the UMC wizard **Active Directory Connection**.

The wizard can be installed from the Univention App Center with the application *Active Directory Connection*. Alternatively, the software package ***univention-ad-connector*** can be installed. Additional information can be found in Section 5.6.

##### Note

All AD and UCS servers in a connector environment must use the same time zone.

##### Warning

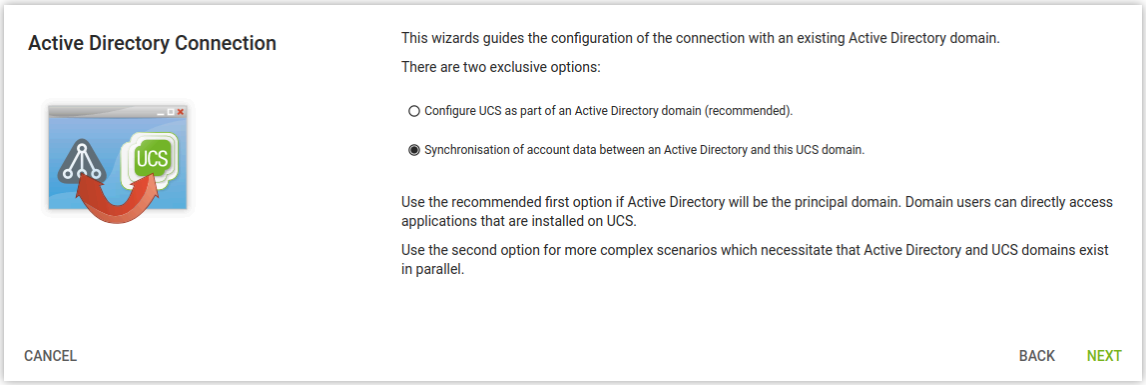
Despite intensive tests it is not possible to rule out that the results of the synchronization may affect the operation of a productive domain. The connector should therefore be tested for the respective requirements in a separate environment in advance.

It is convenient to perform the following steps with a web browser from the AD domain controller, as the files need to be downloaded from the AD domain controller and uploaded into Univention Management Console.

Internet Explorer 6 - which is preinstalled on Windows 2003 systems - is not supported by Univention Management Console. The browser must be updated before continuing.

In the first dialog window of the set-up wizard, the point *Synchronization of content data between an AD and this UCS domain* must be selected and confirmed with **[Next]**.

**Figure 9.7. Configuration of the UCS AD Connector in UMC**




The address of an AD domain controller is requested in the next dialogue window. Here you can specify the IP address of a fully qualified DNS name. If the UCS system is not be able to resolve the computer name of the AD system, the AD DNS server can either be configured as the DNS forwarder under UCS or a DNS host record can be created for the AD system in the UMC DNS management (see Section 11.2.2.3).

Alternatively, a static entry can also be adopted in `/etc/hosts` via Univention Configuration Registry, e.g.

```
ucr set hosts/static/192.0.2.100=w2k8-32.ad.example.com
```

In the **Active Directory account** field, the user is configured which is used for the access on the AD. The setting is saved in the Univention Configuration Registry variable `connector/ad/ldap/binddn`. The replication user must be a member of the `Domain Admins` group in the AD.

The password used for the access must be entered in the **Active Directory password** field. On the UCS system it is only saved locally in a file which only the `root` user can read.

Section 9.3.3.5 describes the steps required if these access data need to be adjusted at a later point in time.

Clicking on **[Next]** prompts the set-up wizard to check the connection to the AD domain controller. If it is not possible to create an SSL/TLS-encrypted connection, a warning is emitted in which you are advised to install a certification authority on the AD domain controller. It is recommended to follow this advice. Following this step, the set-up can be continued by clicking **[Next]** again. If it is still not possible to create an SSL/TLS-encrypted connection, a security query appears asking whether to set up the synchronization without SSL encryption. If this is desired, the set-up can be continued by clicking **[Continue without encryption]**. In this case, the synchronization of the directory data is performed unencrypted.

If the AD domain controller supports SSL/TLS-encrypted connections, the set-up wizard offers **Upload AD root certificate** in the next step. This certificate must be exported from the AD certification authority in advance (see Section 9.3.3.2). In contrast, if this step is skipped, the certificate can also be uploaded via the UMC module at a later point in time and the SSL/TLS encryption enabled (until that point all directory data will, however, be synchronized unencrypted).

The connector can be operated in different modes, which can be selected in the next dialogue window **Configuration of Active Directory domain synchronization**. In addition to bidirectional synchronization, repli-

cation can also be performed in one direction from AD to UCS or from UCS to AD. Once the mode has been selected, **[Next]** needs to be clicked.

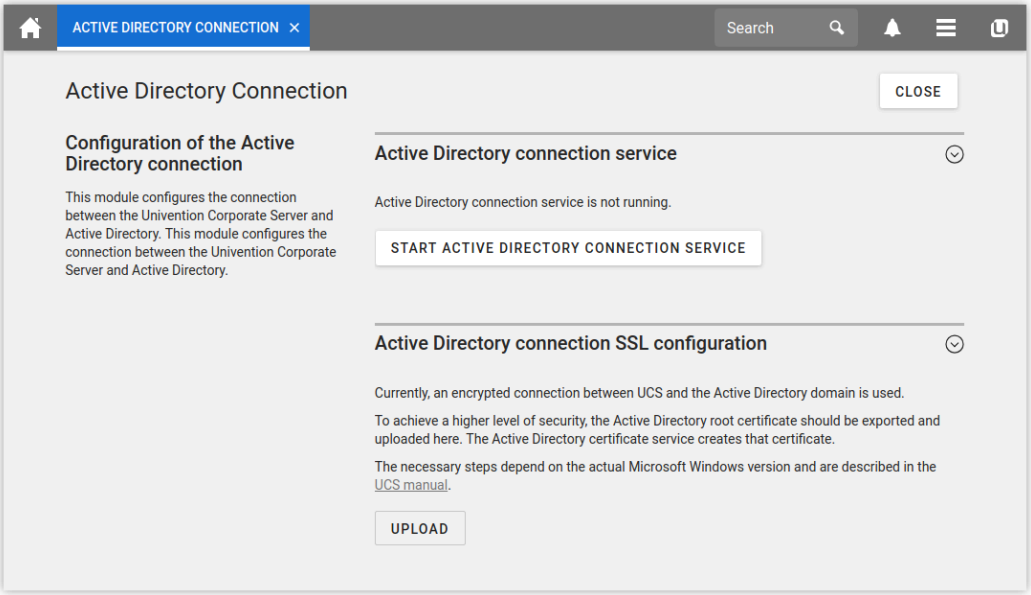
Once **[Next]** is clicked, the configuration is taken over and the UCS AD Connector started. The subsequent dialogue window needs to be closed by clicking on **[Finish]**.

Following this set-up, the *Active Directory Connection* UMC module can be used for further administration of the Active Directory Connection, e.g., for checking whether the service is running and restart it if necessary (see Section 9.3.3.3).

## Note

The connector can also synchronize several AD domains within one UCS domain; this is documented in [ext-doc-win].

**Figure 9.8. Administration dialogue for the Active Directory Connection**




### 9.3.3.2. Importing the SSL certificate of the Active Directory

Feedback

An SSL certificate must be created on the Active Directory system and the root certificate exported to allow encrypted communication. The certificate is created by the Active Directory's certificate service. The necessary steps depend on the Windows versions used. Three versions are shown below as examples.

The encrypted communication between the UCS system and Active Directory can also be deactivated by setting the Univention Configuration Registry variable `connector/ad/ldap/ssl` to `no`. This setting does not affect the replication of encrypted password data.

#### 9.3.3.2.1. Exporting the certificate on Windows 2003

Feedback

The certificate service can be installed subsequently if necessary: **Start -> Properties -> System settings -> Software -> Windows components, choose Certificate Services -> Next select Enterprise root CA -> Next, Enter domain name -> Next -> Next**.

The AD server should be rebooted after the installation.

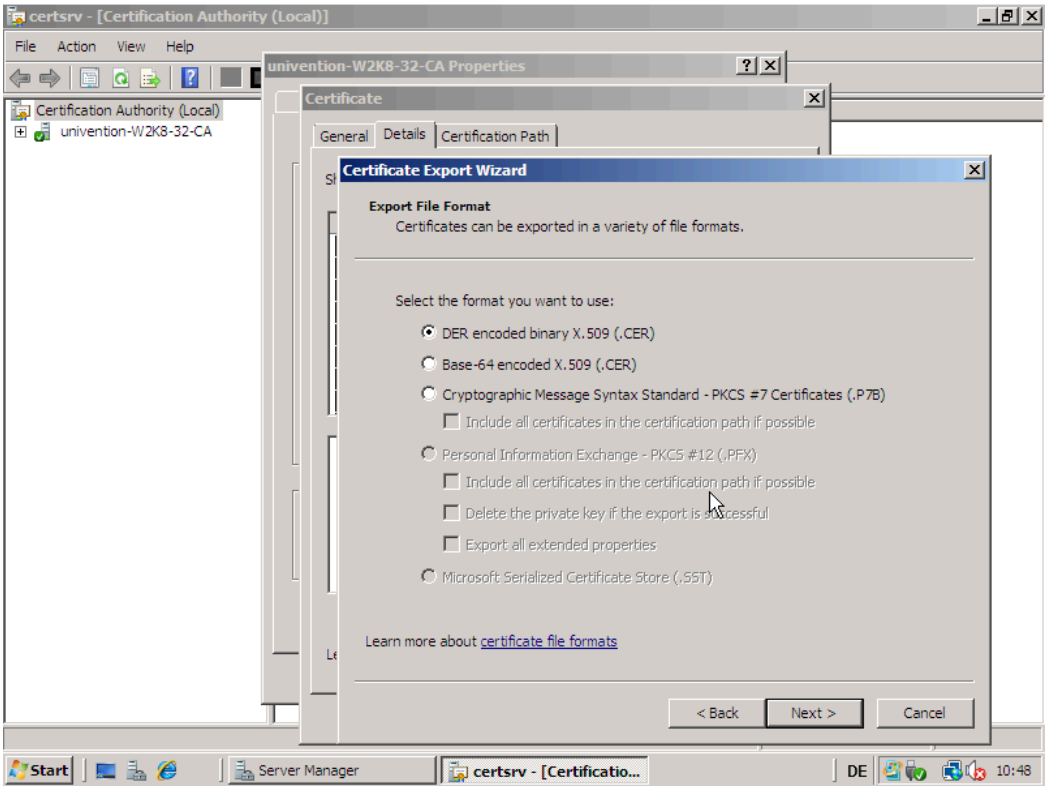
This certificate must now be exported and copied onto the UCS system: **Root CA -> AD domain -> Properties -> Show certificate -> Details -> Copy to file -> DER binary encoded X.509**.

#### 9.3.3.2.2. Exporting the certificate on Windows 2008

Feedback

If the certificate service is not installed, it must be installed before proceeding.

**Figure 9.9. Exporting the root certificate on Windows 2008**




**Start -> Server Manager -> Add or Remove Programs -> Add Roles -> Next -> Active Directory Certificate Services -> Next -> Next -> activate Certification Authority -> select Enterprise -> select Root CA -> Create new private key -> Next -> Accept the proposed crypto setting -> Next -> Accept the proposed name for the CA -> Select an arbitrary validate date -> Next -> Accept default path for the certificate database**.

The following dialogue contains a warning that the name and domain setting cannot be changed again once the certificate authority is installed. This must be confirmed with **Install**.

The AD server must then be restarted.

This certificate must now be exported and copied onto the UCS system: **Start -> Administrative Tools -> Certification -> Authority**. A computer list is shown there and the elements **Revoked Certificates**, **Issued Certificates**, **Pending Requests**, **Failed Requests** and **Certificate Templates** displayed under every system. Here, one must right click on the computer name - not on one of the elements - and then select **Properties**. The root certificate is usually called **Certificate #0**. Then select **Open -> Copy to File -> DER encoded binary X.509 (.CER) -> Select an arbitrary filename -> Finish**.

#### 9.3.3.2.3. Exporting the certificate on Windows 2012

Feedback

If the certificate service is not installed, it must be installed before proceeding.

The server manager must be opened. There, select the **Active Directory Certificate Services** role in the **Manage -> Add Roles and Features** menu. When selecting the role services, it is sufficient simply to select **Certification Authority**. A yellow warning triangle is then shown in the top bar in the server manager. Here, the **Configure Active Directory Certificate Services on the server** option must be selected. **Certification Authority** is selected as the role service to be configured. The type of installation is **Enterprise CA -> Root CA** Now, click on **Create a new private key** and confirm the suggested encryption settings and the suggested name of the certification authority. Any period of validity can be set. The standard paths can be used for the database location.

The AD server must then be restarted.

This certificate must now be exported and copied onto the UCS system: **Server Manager -> Active Directory Certificate Services** Then right click on the server and select **Certification Authority**. There, right click on the name of the generated certificate and **Open -> Copy to File -> DER encoded binary X.509 (.CER) -> Select an arbitrary filename -> Finish**.

A computer list is shown there and the elements **Revoked Certificates**, **Issued Certificates**, **Pending Requests**, **Failed Requests** and **Certificate Templates** displayed under every system. Here, one must right click on the computer name - not on one of the elements - and then select **Properties**. The root certificate is usually called **Certificate #0**. Then select **Open -> Copy to File -> DER encoded binary X.509 (.CER) -> Select an arbitrary filename -> Finish**.

#### 9.3.3.2.4. Copying the Active Directory certificate to the UCS system

Feedback

The SSL AD certificate should now be imported into the UCS system using the UMC wizard.

This is done by clicking on **[Upload]** in the sub menu **Active Directory connection SSL configuration**.

This opens a window in which a file can be selected, which is being uploaded and integrated into the UCS AD Connector.

### 9.3.3.3. Starting/Stopping the Active Directory Connection

Feedback

The connector can be started using **[Start Active Directory connection service]** and stopped using **[Stop Active Directory connection service]**. Alternatively, the starting/stopping can also be performed with the `/etc/init.d/univention-ad-connector` init-script.

### 9.3.3.4. Functional test of basic settings

Feedback

The correct basic configuration of the connector can be checked by searching in Active Directory from the UCS system. Here one can search e.g. for the administrator account in Active Directory with `univention-adsearch cn=Administrator`.

As `univention-adsearch` accesses the configuration saved in Univention Configuration Registry, this allows you to check the reachability/configuration of the Active Directory access.

### 9.3.3.5. Changing the AD access password

Feedback

The access data required by the UCS AD Connector for Active Directory are configured via the Univention Configuration Registry variable `connector/ad/ldap/binddn` and `connector/ad/ldap/bindpw`. If the password has changed or you wish to use another user account, these variables must be adapted manually. The Univention Configuration Registry variable `connector/ad/ldap/binddn` is used to configure the LDAP DN of a privileged replication user. This must be a member of the `Domain Admins` group in the AD. The corresponding password must be saved locally in a file on the UCS system, the name of which must be entered in the Univention Configuration Registry variable `connector/ad/ldap/bind-`

`pw`. The access rights for the file should be restricted so that only the `root` owner has access. The following commands show this as an example:

```
eval "$(ucr shell)"
echo "Updating ${connector_ad_ldap_bindpw?}"
echo "for AD sync user ${connector_ad_ldap_binddn?}"
touch "${connector_ad_ldap_bindpw?}"
chmod 600 "${connector_ad_ldap_bindpw?}"
echo -n "Current AD Syncuser password" > "${connector_ad_ldap_bindpw?}"
```

## 9.3.4. Additional tools / Debugging connector problems

Feedback

The UCS AD Connector provides the following tools and log files for diagnosis:

### 9.3.4.1. `univention-adsearch`

Feedback

This tool facilitates a simple LDAP search in Active Directory. Objects deleted in AD are always shown (they are still kept in an LDAP subtree in AD). As the first parameter the script awaits an LDAP filter; the second parameter can be a list of LDAP attributes to be displayed.

Example:

```
univention-adsearch cn=administrator cn givenName
```

### 9.3.4.2. `univention-connector-list-rejected`

Feedback

This tool lists the DNs of non-synchronized objects. In addition, in so far as temporarily stored, the corresponding DN in the respective other LDAP directory will be displayed. In conclusion *lastUSN* shows the ID of the last change synchronized by AD.

This script may display an error message or an incomplete output if the AD connector is in operation.

### 9.3.4.3. Logfiles

Feedback

For troubleshooting when experiencing synchronization problems, corresponding messages can be found in the following files on the UCS system:

```
/var/log/univention/connector.log
/var/log/univention/connector-status.log
```

## 9.3.5. Details on preconfigured synchronization

Feedback

All containers which are ignored due to corresponding filters are exempted from synchronization as standard. This can be found in the `/etc/univention/connector/ad/mapping` configuration file under the *global_ignore_subtree* setting. To except users from synchronization their user name can be added to the Univention Configuration Registry variable `connector/ad/mapping/user/ignorelist`. For more flexibility a filter can be set in the Univention Configuration Registry variable `connector/ad/mapping/user/ignorefilter`. However this filter does not support the full LDAP filter syntax. It is always case sensitive and the placeholder "*" can only be used as a single value without any other characters.

### 9.3.5.1. Containers and organizational units

Feedback

Containers and organizational units are synchronized together with their description. In addition, the `cn=mail` and `cn=kerberos` containers are ignored on both sides. Some particularities must be noted for

containers on the AD side. In the **User manager** Active Directory offers no possibility to create containers, but displays them only in the advanced mode (**View -> Advanced settings**).

#### 9.3.5.1.1. Particularities

Feedback

- Containers or organizational units deleted in AD are deleted recursively in UCS, which means that any non-synchronized subordinate objects, which are not visible in AD, are also deleted.

### 9.3.5.2. Groups

Feedback

Groups are synchronized using the group name, whereby a user's primary group is taken into account (which is only stored for the user in LDAP in AD).

Group members with no opposite in the other system, e.g., due to ignore filters, are ignored (thus remain members of the group).

The description of the group is also synchronized.

#### 9.3.5.2.1. Particularities

Feedback

- The *pre Windows 2000 name* (LDAP attribute *samAccountName*) is used in AD, which means that a group in Active Directory can appear under a different name from in UCS.
- The connector ignores groups, which have been configured as a *Well-Known Group* under **Samba group type** in Univention Directory Manager. There is no synchronization of the SID or the RID.
- Groups which were configured as *Local Group* under **Samba group type** in Univention Directory Manager are synchronized as a *global group* in the Active Directory by the connector.
- Newly created or moved groups are always saved in the same subcontainer on the opposite side. If several groups with the same name are present in different containers during initialization, the members are synchronized, but not the position in LDAP. If one of these groups is migrated on one side, the target container on the other side is identical, so that the DNs of the groups can no longer be differentiated from this point onward.
- Certain group names are converted using a mapping table so that, for example in a German language setup, the UCS group `Domain Users` is synchronized with the AD group *Domänen-Benutzer*. When used in anglophone AD domains, this mapping can result in *germanophone* groups' being created and should thus be deactivated in this case. This can be done using the Univention Configuration Registry variable `connector/ad/mapping/group/language`

  The complete table is:

| ***UCS group*** | ***AD group*** |
|---|---|
| `Domain Users` | `Domänen-Benutzer` |
| `Domain Admins` | `Domänen-Admins` |
| `Windows Hosts` | `Domänencomputer` |

- Nested groups are represented differently in AD and UCS. In UCS, if groups are members of groups, these objects can not always be synchronized on the AD side and appear in the list of rejected objects. Due to the existing limitations in Active Directory, nested groups should only be assigned there.
- If a global group `A` is accepted as a member of another global group `B` in Univention Directory Manager, this membership does not appear in Active Directory because of the internal AD limitations in `Windows 2000/2003`. If group `A`'s name is then changed, the group membership to group `B` will be lost. Since

`Windows 2008` this limitation no longer exists and thus global groups can also be nested in Active Directory.

#### 9.3.5.3. Users

Feedback

Users are synchronized like groups using the user name or using the AD pre Windows 2000 name. The *First name*, *Last name*, *Primary group* (in so far as present on the other side), *Organization*, *Description*, *Street*, *City*, *Postal code*, *Windows home path*, *Windows login script*, *Disabled* and *Account expiry date* attributes are transferred. Indirectly *Password*, *Password expiry date* and *Change password on next login* are also synchronized. *Primary e-mail address* and *Telephone number* are prepared but commented out due to differing syntax in the mapping configuration.

The `root` and `Administrator` users are exempted.

##### 9.3.5.3.1. Particularities

Feedback

- Users are also identified using the name, so that for users created before the first synchronization on both sides, the same process applies as for groups as regards the position in LDAP.
- In some cases, a user to be created under AD, for which the password has been rejected, is deleted from AD immediately after creation. The reasoning behind this is that AD created this user firstly and then deletes it immediately once the password is rejected. If these operations are transmitted to UCS, they are transmitted back to AD. If the user is re-entered on the AD side before the operation is transmitted back, it is deleted after the transmission. The occurrence of this process is dependent on the polling interval set for the connector.
- AD and UCS create new users in a specific primary group (usually `Domain Users` or `Domänen-Benutzer`) depending on the presetting. During the first synchronization from UCS to AD the users are therefore always a member in this group.

## 9.4. Migrating an Active Directory domain to UCS using Univention AD Takeover

Feedback

### 9.4.1. Introduction

Feedback

UCS supports the takeover of user, group and computer objects as well as Group Policy Objects (GPOs) from a Microsoft Active Directory (AD) domain. Windows clients do not need to rejoin the domain. The takeover is an interactive process consisting of three distinct phases:

- Copying all objects from Active Directory to UCS
- Copying of the group policy files from the AD server to UCS
- Deactivation of the AD server and assignment of all FSMO roles to the UCS DC

The following requirements must be met for the takeover:

- The UCS domain controller (master domain controller) needs to be installed with a unique hostname, not used in the AD domain.
- The UCS domain controller needs to be installed with the same DNS domain name, NetBIOS (pre Windows 2000) domain name and Kerberos realm as the AD domain. It is also recommended to configure the same LDAP base DN.
- The UCS domain controller needs to be installed with a unique IPv4 address in the same IP subnet as the Active Directory domain controller that is used for the takeover.

### Caution

If the system is already a member of an Active Directory Domain, installing the *Active Directory Takeover* application removes this membership. Therefore, the installation of the *Takeover* application has to take place only shortly before the actual takeover of the AD domain.

The *Active Directory Takeover* application must be installed from the Univention App Center for the migration. It must be installed on the system where the Univention S4 Connector is running (see Section 9.2.2.4, usually the master domain controller).

## 9.4.2. Preparation

Feedback

The following steps are strongly recommended before attempting the takeover:

- A backup of the AD server(s) should be performed.
- If user logins to the AD server are possible (e.g. through domain logins or terminal server sessions) it is recommended to deactivate them and to stop any services in the AD domain, which deliver data, e.g. mail servers. This ensures that no data is lost in case of a rollback to the original snapshot/backup.
- It is recommended to set the same password for the `Administrator` account on the AD server as the corresponding account in the UCS domain. In case different passwords are used, the password that was set last, will be the one that is finally valid after the takeover process (timestamps are compared for this).
- In a default installation the `Administrator` account of the AD server is deactivated. It should be activated in the local user management module.

The activation of the `Administrator` account on the AD server is recommended because this account has all the required privileges to copy the GPO SYSVOL files. The activation can be achieved by means of the **Active Directory Users and Computers** module or by running the following two commands:

```
net user administrator /active:yes
net user administrator PASSWORD
```

## 9.4.3. Domain migration

Feedback

The takeover must be initiated on the UCS domain controller that runs the Univention S4 Connector (by default the master domain controller). During the takeover process Samba must only run on this UCS system. If other Samba domain controllers have been added to the UCS domain, they need to be stopped! This is important to avoid data corruption by mixing directory data taken over from Active Directory with Samba 4 directory data replicated from other UCS domain controllers.

Other Samba systems can be stopped by logging into each of the other UCS domain controllers as the `root` user and running

```
/etc/init.d/samba4 stop
```

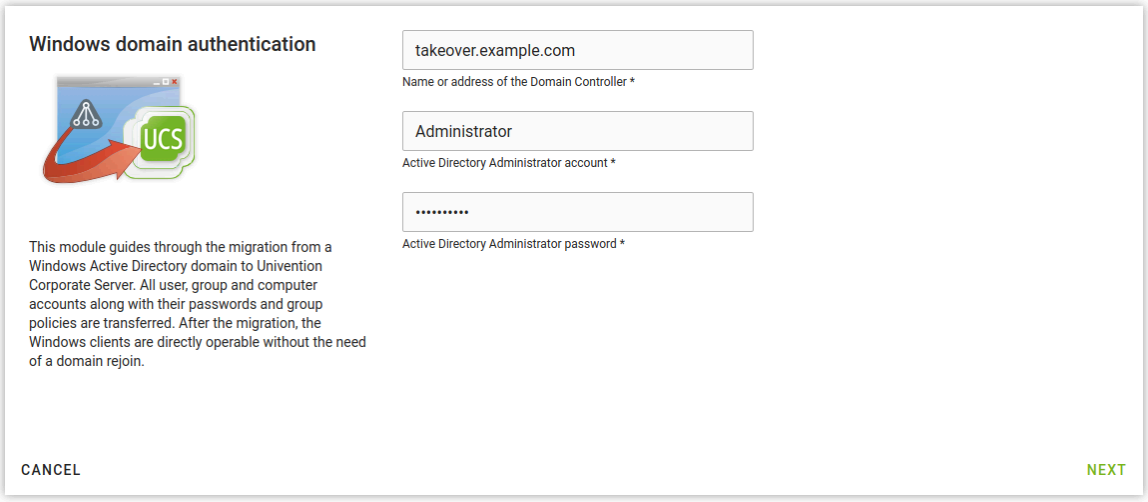
After ensuring that only the Univention S4 Connector host runs Samba 4, the takeover process can be started. If the UCS domain was installed initially with a UCS version before UCS 3.2, the following Univention Configuration Registry variable needs to be set first:

```
ucr set connector/s4/mapping/group/grouptype=false
```

The takeover is performed with the **Active Directory Takeover** Univention Management Console module. The IP address of the AD system must be specified under **Name or address of the Domain Controller**. An

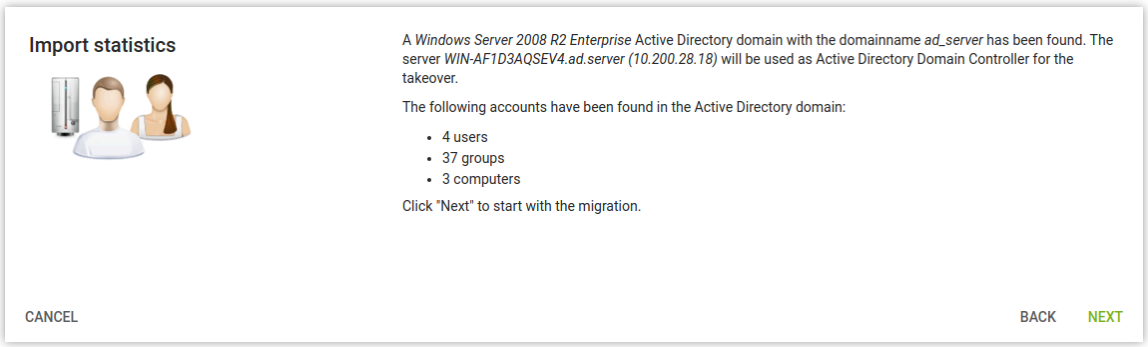
account from the AD domain must be specified under **Active Directory Administrator account** which is a member of the AD group `Domain Admins` (e.g., the `Administrator`) and the corresponding password entered under **Active Directory Administrator password**.

**Figure 9.10. First phase of domain migration**




The module checks whether the AD domain controller can be accessed and displays the domain data to be migrated.

**Figure 9.11. Overview of the data to be migrated**




When **Next** is clicked, the following steps are performed automatically. Additional information is logged to `/var/log/univention/ad-takeover.log` as well as to `/var/log/univention/management-console-module-adtakeover.log`.

- Adjust the system time of the UCS system to the system time of the Active Directory domain controller in case the UCS time is behind by more than three minutes.
- Join the UCS domain controller into the Active Directory domain
- Start Samba and the Univention S4 connector to replicate the Active Directory objects into the UCS OpenLDAP directory
- When "*Well Known*" account and group objects (identified by their special RIDs) are synchronized into the UCS OpenLDAP, a listener module running on each UCS system sets a Univention Configuration Registry variable to locally to map the English name to the non-English AD name. These variables are used to trans-

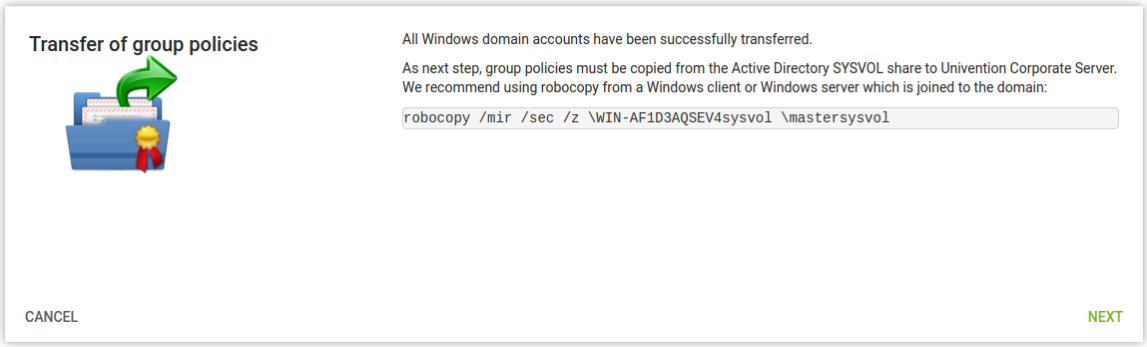
late the English names used in the UCS configuration files to the specific names used in Active Directory. To give an example, if `Domain Admins` has a different name in the AD, then the Univention Configuration Registry variable `groups/default/domainadmins` is set to that specific name (likewise for uses, e.g. `users/default/administrator`).

The UCS domain controller now contains all users, groups and computers of the Active Directory domain. In the next step, the SYSVOL share is copied, in which among other things the group policies are stored.

This phase requires to log onto the Active Directory domain controller as the `Administrator` (or the equivalent non-English name). There a command needs to be started to copy the group policy files from the Active Directory SYSVOL share to the UCS SYSVOL share.

The command to be run in shown in the UMC module. If it has been successfully run, it must be confirmed with **Next**.

**Figure 9.12. Copying the SYSVOL share**

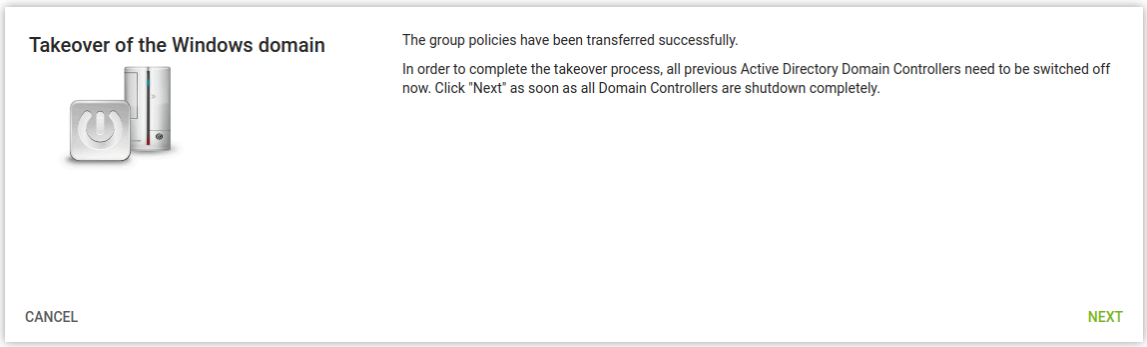



It may be necessary to install the required `robocopy` tool, which is part of the Windows Server 2003 Resource Kit Tools. Starting with Windows 2008 the tool is already installed.

Note: The `/mir` option of `robocopy` mirrors the specified source directory to the destination directory. Please be aware that if you delete data in the source directory and execute this command a second time, this data will also be deleted in the destination directory.

After successful completion of this step, it is now necessary to shutdown all domain controllers of the Active Directory domain. Then **Next** must be clicked in the UMC module.

**Figure 9.13. Shutdown of the AD server(s)**




The following steps are now automatically performed:

- Claiming all FSMO roles for the UCS domain controller. These describe different tasks that a server can take on in an AD domain.
- Register the name of the Active Directory domain controller as a DNS alias (see Section 11.2.2.2) for the UCS DNS server.
- Configure the IP address of the Active Directory domain controller as a virtual Ethernet interface
- Perform some cleanup, e.g. removal of the AD domain controller account and related objects in the Samba SAM account database.
- Finally restart Samba and the DNS server

### 9.4.4. Final steps of the takeover

Feedback

Finally the following steps are required:

- The domain function level of the migrated Active Directory domain needs to be checked by running the following command:

```
samba-tool domain level show
```

  In case this command returns the message **ATTENTION: You run SAMBA 4 on a forest function level lower than Windows 2000 (Native).** the following commands should be run to fix this:

```
samba-tool domain level raise --forest-level=2003 --domain-level=2003
samba-tool dbcheck --fix --yes
```

- In case there has been more than one Active Directory domain controller in the original Active Directory domain, all the host accounts of the other domain controllers must be removed in the computers management module of the Univention Management Console. In addition their accounts must be removed from the Samba SAM database. This may be done by logging on to a migrated Windows client as member of the group `Domain Admins` and running the tool **Active Directory Users and Computers**.
- If more than one UCS domain controller with Samba domain controller has been installed, these servers need to be re-joined.
- All Windows clients need to be rebooted.

### 9.4.5. Tests

Feedback

It is recommended to perform thorough tests with Windows client systems, e.g.

- Login to a migrated client as a migrated user
- Login to a migrated client as the Administrator
- Testing group policies
- Join of a new Windows client
- Creation of a new UCS user and login to a Windows client

## 9.5. Trust relationships

Feedback

Trust relationships between domains make it possible for users from one domain to log on to computers from another domain.

In general, Windows trust relations can be unidirectional or bidirectional. Technically a bidirectional trust is simply realized as two unidirectional trusts, one in each direction.

The terminology of unidirectional trusts depends on the perspective of either the trusting or trusted domain: From the perspective of the trusting domain, the trust is called *outgoing*. From the perspective of the trusted domain, the trust is called *incoming*.

In UCS, outgoing trust (UCS trusts Windows) is not supported currently. As a consequence, bidirectional trust is not supported either.

When setting up and using the trust relationship the domain controllers of both domains must be able to reach each other over the network and identify each other via DNS. At least the fully qualified DNS names of the domain controllers of the respective remote domain must be resolvable to allow communication between both domains to work. This can be achieved by configuring conditional DNS forwarding in both domains.

The following example assumes, that the UCS Samba/AD DC Master `master.ucsdom.example` has the IP address `192.0.2.10` and that the Active Directory domain controller `dc1.addom.example` of the remote domain has the IP address `192.0.2.20`.

On the UCS side the conditional forwarding of DNS queries can be set up as `root` with the following commands:

```
cat >> /etc/bind/local.conf.samba4 <<%EOR

zone "addom.example" {
type forward;
forwarders { 192.0.2.20; };
};
%EOR
service bind9 restart
```

The success can be checked by running `host dc1.addom.example`.

In addition, it may be useful to create a static entry for the domain controller of the remote Active Directory domain in the file `/etc/hosts`:

```
ucr set hosts/static/192.0.2.20=dc1.addom.example
```

On a Windows AD DC, a so-called *conditional forwarding* can be set up for the UCS domain via the DNS server console.

After this preliminary work, the trust itself can be established directly from the command line of the UCS Samba/AD DC.

Trust relationships can only be configured on domain controllers but they affect the whole domain.

In Samba/AD domain the trust relationship can be configured easily on the command line using the tool `samba-tool`:

```
samba-tool domain trust create addom.example \
           -k no -UADDOM\\Administrator%ADAdminPassword \
           --type=external --direction=incoming
```

The trust can be checked using the following commands:

```
samba-tool domain trust list
wbinfo --ping-dc –domain=addom.example
```

```
wbinfo --check-secret –domain=addom.example
```

After the setup, a UCS user should be able to log on to systems of the remote Active Directory domain. Users must either use the format `UCSDOM\username` as login name or their Kerberos principal in the notation `username@ucsdom.example`.

# Chapter 10. Identity Management connection to cloud services

10.1. Introduction ..... 195
10.2. Microsoft 365 Connector ..... 195
10.2.1. Setup ..... 195
10.2.2. Configuration ..... 196
10.2.2.1. Users ..... 196
10.2.2.2. Teams ..... 197
10.2.3. Synchronization of Users in multiple Azure Active Directories ..... 197
10.2.4. Troubleshooting/Debugging ..... 198
10.3. Google Apps for Work Connector ..... 198
10.3.1. Setup ..... 198
10.3.2. Configuration ..... 199
10.3.3. Troubleshooting/Debugging ..... 200

## 10.1. Introduction

Feedback

UCS offers an integrated Identity Management System. Through Univention Management Console, users and groups among others can easily be administered. Depending on the installed services, these identities are made available through different interfaces e.g. via LDAP.

The management system can be extended with the help of provided extensions, also called Apps. Thus users or groups can also be replicated in cloud services. In the App center there are also among others extensions for Microsoft 365 or G Suite.

Thanks to Single Sign-On (SSO), users can log in with their usual password and immediately get to work online in the cloud. The password remains in the company's network and is not transferred to the cloud service.

The following chapter describes how to set up the Microsoft 365 Connector and the Google Apps for Work Connector.

## 10.2. Microsoft 365 Connector

Feedback

The synchronization of users and groups and teams to an Azure Directory Domain, which will then be used by Microsoft 365, is made possible by the Microsoft Office connector. The connector makes it possible to control which of the users created in UCS can use Microsoft 365. The selected users will be provisioned accordingly into the Azure Active Directory domain. It is configurable which user attributes are synchronized and which are anonymized during synchronization.

The Single Sign-On log-in to Microsoft 365 is done via the UCS integrated SAML implementation. Authentication takes place against the UCS server, and no password hashes are transmitted to Microsoft Azure Cloud. The user's authentication is done exclusively via the client's web browser. The web browser should however be able to resolve the DNS records of the UCS domain, this is a particularly important point to note for mobile devices.

### 10.2.1. Setup

Feedback

To setup the Microsoft 365 Connector a Microsoft 365 Administrator account, a corresponding Account in the Azure Active Directory, as well as a  Domain verified by Microsoft [1] are required. The first two are provided

[1] https://azure.microsoft.com/en-us/documentation/articles/active-directory-add-domain/

for test purposes by Microsoft for free. However to configure the SSO, a separate Internet domain where TXT records can be created is required.

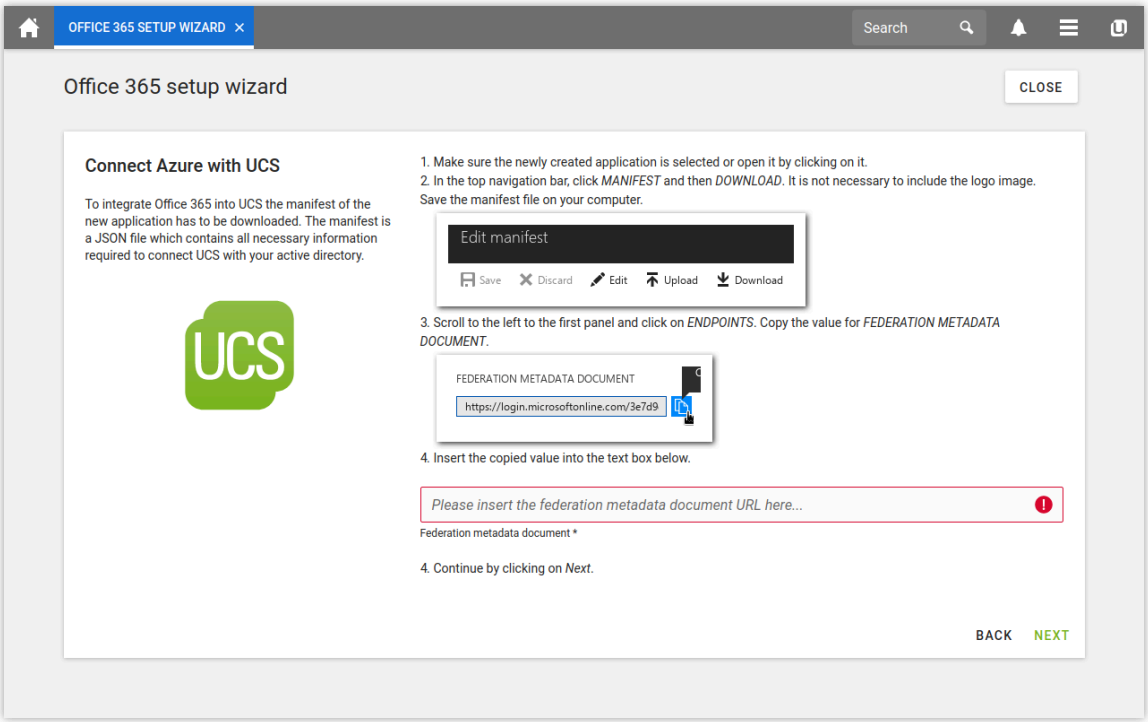
In case there is no Microsoft 365 subscription available, one can be configured it via https://www.office.com/ in the **trial for business** section. A connection is not possible with a private Microsoft account.

You should then log in with a **Microsoft 365 Administrator Account** into the **Microsoft 365 Admin Center**. At the bottom left of the navigation bar select **Azure AD** to open the **Azure Management Portal** in a new window.

In the **Azure Active Directory** section the menu item **Custom domain names** can be used to add and verify your own domain. For this it is necessary to create a TXT record in the DNS of your own domain. This process can take up to several minutes. Afterwards the **status** of the configured domain will be displayed as **Verified**.

Now the Microsoft 365 App can be installed from the App Center on the UCS system. The installation takes a few minutes. There is a setup wizard available for the initial configuration. After completing the wizard the connector is ready for use.

**Figure 10.1. Microsoft 365 Setup assistant**




## 10.2.2. Configuration

Feedback

After the end of the installation through the setup wizard, users can be enabled to use Microsoft 365. This configuration can be done through the user module on each user object on the **Microsoft 365** tab. Usage and allocation of licenses are acknowledged in the **Microsoft 365 Admin Center**.

### 10.2.2.1. Users

Feedback

If a change is made to the user, the changes are likewise replicated to the Azure Active Directory domain. There is no synchronization from the Azure Active Directory to the UCS system. This means changes made in Azure Active Directory or Office Portal may be overridden by changes to the same attributes in UCS.

Due to Azure Active Directory security policies, users or groups in the Azure AD can't be deleted during synchronization. They are merely disabled and renamed. The licenses are revoked in the Azure Active Directory so that they become available to other users. Users and groups whose names start with **ZZZ_deleted** can be deleted in **Microsoft 365 Admin Center**.

It is necessary to configure a country for the user in Microsoft 365. The connector uses the specification of the Country from the contact data of the user. If not set, it uses the setting of the server. With the help of Univention Configuration Registry-Variable `office365/attributes/usageLocation` a 2-character abbreviation, e.g. US, can be set as the default.

Through Univention Configuration Registry-Variable `office365/attributes/sync`, the LDAP attributes (e.g. first name, last name, etc.) of a user's account which will to be synchronized are configured. The form is a comma-separated list of LDAP attributes. Thus adaptation to personal needs is easily possible.

With the Univention Configuration Registry-Variable`office365/attributes/anonymize`, a comma-separated list of LDAP attributes can be configured that are created in the Azure Active Directory but filled with random values. The Univention Configuration Registry-Variables `office365/attributes/static/.*` allows the filling of attributes on the Microsoft side with a predefined value.

The Univention Configuration Registry-Variable`office365/attributes/never` can be used to specify a comma separated list of LDAP attributes that should not be synchronized even when they appear in `office365/attributes/sync` or `office365/attributes/anonymize`.

The Univention Configuration Registry-Variables `office365/attributes/mapping/.*` define a mapping of UCS LDAP attributes to Azure Attributes. Usually these variables don't need to be changed. The synchronization of the groups of Microsoft 365 user can be enabled with the Univention Configuration Registry-Variable`office365/groups/sync`.

Changes to Univention Configuration Registry-Variables are implemented only after restarting the Univention Directory Listener.

#### 10.2.2.2. Teams

Feedback

To use Teams, synchronization of groups must be enabled via Univention Configuration Registry-Variable `office365/groups/sync=yes`, and then the Univention Directory Listener service must be restarted. If UCS groups are to be created as teams in Microsoft 365, the groups must be configured as teams on the **Microsoft 365** tab via the **Microsoft 365 Team** checkbox. Furthermore, it is necessary to define an owner of the team on the same tab. Further settings on the team can be made by the team owners directly in the Teams interface. After activating a group as a team, the group members are added to the new team. Provisioning a new team in Microsoft 365 can take a few minutes.

Ensure that users of a team in Azure are provided with a license that includes the use of Teams.

### 10.2.3. Synchronization of Users in multiple Azure Active Directories

Feedback

The Microsoft 365 Connector is able to synchronize users to multiple Azure Active Directories. For each user account, multiple Azure AD instances can be assigned, where an account should be created. A user gets a distinct username (*Userprincipalname*, UPN) for every assigned Azure AD.

An alias is assigned to each additional Azure AD connection by the administrator. To manage these aliases the program `/usr/share/univention-office365/scripts/manage_adconnections` can be used. A new alias is created by calling `/usr/share/univention-office365/scripts/manage_adconnections create <Aliasname>`. This will set the Univention Configuration Registry-Variable `office365/adconnection/wizard` to the newly created alias. The value of this Univention Configuration Registry-Variable defines which connection is configured by the next run of the Microsoft 365 Configuration Wizard.

After creating the alias, the new connection must be configured through the Microsoft 365 Configuration Wizard as well.

To use Single Sign-On with multiple Azure AD connections, a new logical SAML Identity Provider is needed for each connection. This is described in Section 3.9.3. The Identity Provider should get the same name as the alias. If another name was chosen, the PowerShell script to configure Single Sign-On needs to be adjusted manually. For example the Univention Configuration Registry-Variable `saml/idp/entityID/supplement/Aliasname=true` needs to be set on all domain controllers responsible for Single Sign-On.

A UCS user can only use one Microsoft 365 account in one browser session at a time. To change the connection, a logout from Microsoft 365 is necessary.

A default alias for Microsoft 365 enabled users and groups can be set in the Univention Configuration Registry-Variable `office365/defaultalias`. To synchronize them into a different Azure Active Directory the connection alias must be selected explicitly at the user or group object.

### 10.2.4. Troubleshooting/Debugging

Feedback

Messages during the setup are logged in `/var/log/univention/management-console-module-office365.log`.

In case of synchronization problems, the log file of the Univention Directory Listener should be examined: `/var/log/univention/listener.log` Some actions of the Connector use long-running Azure Cloud operations, especially when using Teams. These operations are logged in the log file `/var/log/univention/listener_modules/ms-office-async.log` The Univention Configuration Registry-Variable`Office365/debug/werror` activates additional debug output.

## 10.3. Google Apps for Work Connector

Feedback

Google Apps for Work Connector allows users and groups to synchronize to a G Suite domain. You can control which of the users created in UCS are allowed to use G Suite. The users selected in this way are provisioned accordingly by UCS into the G Suite domain. It can be configured which attributes are synchronized and attributes can be anonymized.

The Single Sign-On log-in to G Suite is done via the UCS integrated SAML implementation. Authentication takes place against the UCS server, and no password hashes are transferred to the G Suite domain. The user's authentication is done exclusively via the client's web browser. However, the browser should be able to resolve the DNS records of the UCS domain, which is particularly important for mobile devices.

### 10.3.1. Setup

Feedback

To setup the Google Apps for Work Connectors a G Suite Administrator account, a corresponding account in the G Suite domain, and a domain verified[2] by Google are required. The first two will be provided free of charge by Google for testing purposes. However, configuring the SSO requires a separate Internet domain where TXT records can be created.

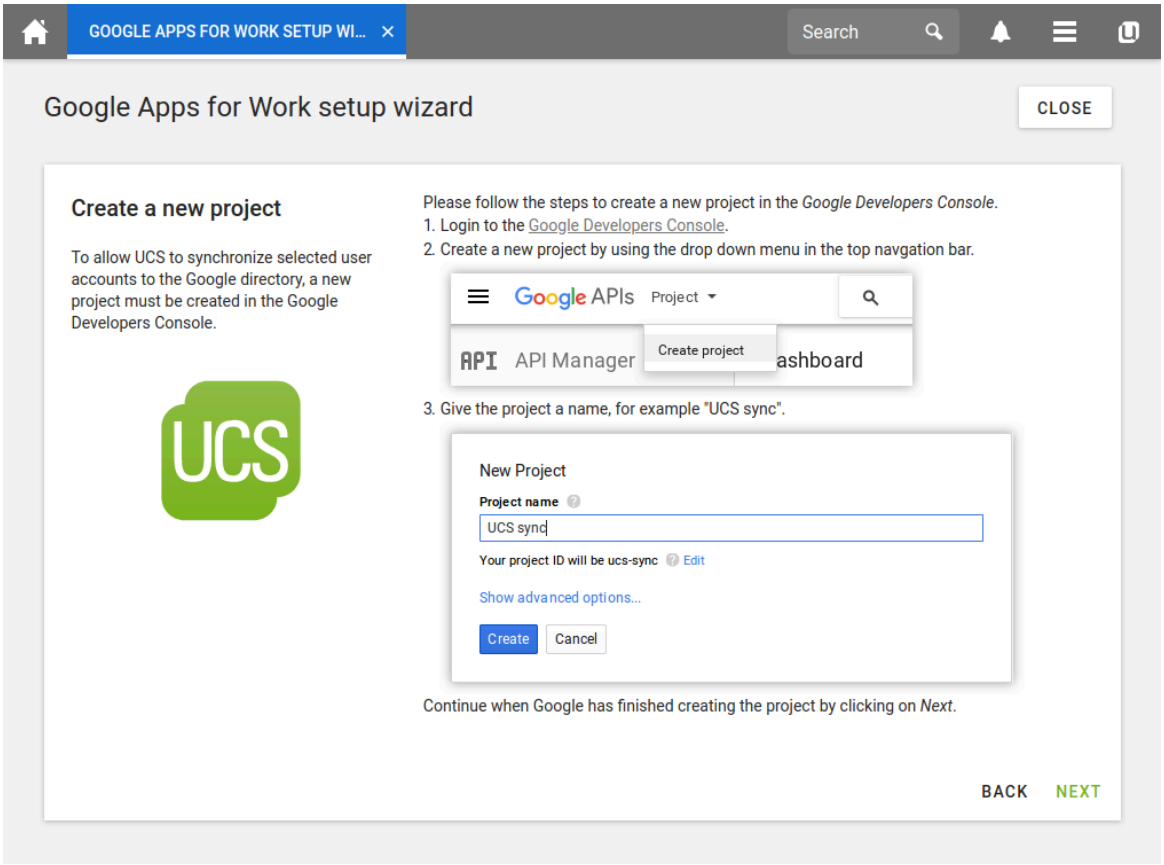
If no G Suite subscription is available yet, it can be configured via https://gsuite.google.com/setup-hub/ with the link **Start your free trial**. A connection with a private Gmail account is not possible.

Afterwards, you should log in with a **G Suite administrator account** in the Admin Console[3]. The domain should now be verified. For this it is necessary to create a TXT record in the DNS of your own domain. This process can take a few minutes.

[2] https://support.google.com/a/topic/9196?hl=en
[3] https://admin.google.com/

Now the Google Apps for Work Connector from the App Center can be installed on the UCS system. The installation only takes a few minutes. There is a setup wizard available for the initial configuration. After completing the wizard the connector is ready for use.

**Figure 10.2. Google Apps for Work Setup Wizard**




## 10.3.2. Configuration

Feedback

After the setup via the setup wizard, you can use the user module on each user object on the **Google Apps tab** to configure that this user is provisioned to G Suite.

If a change is made to the user, the changes will also be replicated to the G Suite domain. There is no synchronization from the G Suite domain to the UCS system. This means that changes made in the G Suite domain may be overwritten by changes to the same attributes in UCS.

If the Google Apps property is removed from a user, the user will be deleted from the G Suite domain accordingly.

The Univention Configuration Registry-Variable `google-apps/attributes/mapping/.*` is used to configure which LDAP attributes (e.g. first name, last name, etc.) of a user account are synchronized. The Univention Configuration Registry-Variable and its values reflect the nested data structure of the G Suite user accounts. The names that follow the percentage sign in the values are the attributes in the UCS LDAP. If all Univention Configuration Registry-Variable `google-apps/attributes/mapping/.*` are removed, no data other than the primary e-mail address is synchronized.

The Univention Configuration Registry-Variable `google-apps/attributes/anonymize` can be used to specify comma-separated LDAP attributes that are created in the G Suite domain but filled with random values.

The Univention Configuration Registry-Variable `google-apps/attributes/never` can be used to specify comma-separated LDAP attributes that should not be synchronized, even if they are configured via `google-apps/attributes/mapping` or `google-apps/attributes/anonymize`.

The synchronization of Google Apps for Work user groups can be enabled with the Univention Configuration Registry-Variable `google-apps/groups/sync`.

Changes to Univention Configuration Registry-Variable are implemented after restarting the Univention Directory Listener.

## 10.3.3. Troubleshooting/Debugging

Feedback

Messages during setup are logged in the following log file `/var/log/univention/management-console-module-googleapps.log`.

In case of synchronization problems, the log file of the Univention Directory Listener should be checked: `/var/log/univention/listener.log`. The Univention Configuration Registry-Variable `google-apps/debug/werror` activates additional debug output.

univention
be open.

# Chapter 11. IP and network management


11.1. Network objects ..... 202
11.2. Administration of DNS data with BIND ..... 203
11.2.1. Configuration of the BIND name server ..... 204
11.2.1.1. Configuration of BIND debug output ..... 204
11.2.1.2. Configuration of the data backend ..... 204
11.2.1.3. Configuration of zone transfers ..... 205
11.2.2. Administration of DNS data in Univention Management Console ..... 205
11.2.2.1. Forward lookup zone ..... 205
11.2.2.2. CNAME record (Alias records) ..... 207
11.2.2.3. A/AAAA records (host records) ..... 207
11.2.2.4. Service records ..... 208
11.2.2.5. Reverse lookup zone ..... 209
11.2.2.6. Pointer record ..... 209
11.3. IP assignment via DHCP ..... 210
11.3.1. Introduction ..... 210
11.3.2. Composition of the DHCP configuration via DHCP LDAP objects ..... 211
11.3.2.1. Administration of DHCP services ..... 211
11.3.2.2. Administration of DHCP server entries ..... 211
11.3.2.3. Administration of DHCP subnets ..... 212
11.3.2.4. Administration of DHCP pools ..... 212
11.3.2.5. Registration of computers with DHCP computer objects ..... 213
11.3.2.6. Management of DHCP shared networks / DHCP shared subnets ..... 214
11.3.3. Configuration of clients via DHCP policies ..... 214
11.3.3.1. Setting the gateway ..... 215
11.3.3.2. Setting the DNS servers ..... 215
11.3.3.3. Setting the WINS server ..... 215
11.3.3.4. Configuration of the DHCP lease ..... 216
11.3.3.5. Configuration of boot server/PXE settings ..... 216
11.3.3.6. Further DHCP policies ..... 216
11.4. Packet filter with Univention Firewall ..... 217
11.5. Web proxy for caching and policy management / virus scan ..... 217
11.5.1. Installation ..... 217
11.5.2. Caching of web content ..... 218
11.5.3. Logging proxy accesses ..... 218
11.5.4. Restriction of access to permitted networks ..... 218
11.5.5. Configuration of the ports used ..... 218
11.5.5.1. Access port ..... 218
11.5.5.2. Permitted ports ..... 219
11.5.6. User authentication on the proxy ..... 219
11.5.7. Filtering/policy enforcement of web content with DansGuardian ..... 219
11.5.8. Definition of content filters for DansGuardian ..... 220
11.6. RADIUS ..... 222
11.6.1. Installation ..... 222
11.6.2. Configuration ..... 222
11.6.2.1. Allowed users ..... 222
11.6.2.2. MAC filtering ..... 222
11.6.2.3. Access point administration ..... 223
11.6.2.4. Access point and client configuration ..... 224
11.6.3. Debugging ..... 224


This chapter describes how IP addresses for computer systems in a UCS domain can be centrally managed via Univention Management Console and and assigned via DHCP.

*Network objects* (Section 11.1) bundle available IP address segments of a network. The DNS resolution as well as the assignment of IP addresses via DHCP are integrated in UCS, as detailed in Section 11.2 and Section 11.3.

Incoming and outgoing network traffic can be restricted via the *Univention Firewall* based on `iptable` (Section 11.4).

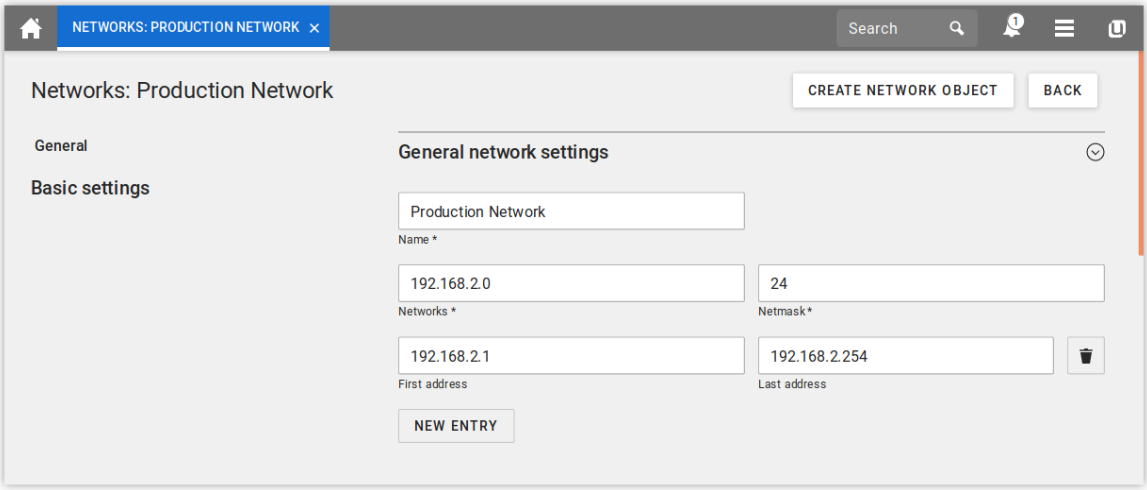
The integration of the proxy server Squid allows the caching of web contents and the enforcement of content policies for web access (Section 11.5).

# 11.1. Network objects

Feedback

*Network objects* can be used to compile available IP addresses; the next available address is then automatically specified during assignment to a computer.

### Figure 11.1. Creating a network object




For example, it is possible to define a network object *Workstation network* which encompasses the IP addresses from `192.0.2.0` to `192.0.2.254`. If a Windows computer object is now created and only the network object selected, an internal check is performed for which IP addresses are already assigned and the next free one selected. This saves the administrator having to compile the available addresses manually. If a computer object is removed, the address is automatically reassigned.

Network objects are managed in the UMC module *Networks* (see Section 4.4)

### Table 11.1. 'General' tab

| Attribute | Description |
|---|---|
| Name | The name of the network is entered in this input field. This is the name under which the network also appears in the computer management. |
| Networks | The network address is entered in dot-decimal form in this input field, e.g., `192.0.2.0`. |
| Netmask | The network mask can be entered in this input field in network prefix or dot-decimal form. If the network mask is entered in dot-decimal form it will be subsequently be converted into the corresponding network prefix and later also shown so. |

| Attribute | Description |
|---|---|
| IP address range | One or more IP ranges can be configured here. When a host is assigned to this network at a later point, it will automatically be assigned the next, free IP address from the IP range entered here.<br><br>When no IP range is entered here, the system automatically uses the range given by the network and the subnet mark entered.<br><br>Forward lookup zones and reverse lookup zones can be selected in the sub menu **DNS preferences**. When a host is assigned to this network at a later point, a host record in the forward lookup zone and/or a pointer record in the reverse lookup zone will be created automatically.<br><br>The zones are also administrated in Univention Management Console, see Section 11.2.2.1.<br><br>If no zone is selected here, no DNS records are created during assignment to a computer object. However, the DNS entries can still be set manually. |
| DNS forward lookup zone | The forward lookup zone where hosts from the network should be added must be specified here. The resolution of the computer name to an IP address is performed via the zone. |
| DNS reverse lookup zone | The reverse lookup zone where hosts from the network should be added must be specified here. The reverse resolution of the IP address back to a computer name is performed via the zone.<br><br>A DHCP service can be assigned to the network in the sub menu **DHCP preferences**. When a host is assigned to this network at a later point, a DHCP computer entry with a fixed IP address will be created automatically in the selected DHCP service.<br><br>The DHCP service settings are also administrated in Univention Management Console, see Section 11.3.2.<br><br>If no DHCP service is selected, no DHCP host record is created during assignment to a computer object. However, such an entry can also still be assigned manually. |

## 11.2. Administration of DNS data with BIND

Feedback

UCS integrates BIND for the name resolution via the domain name system (DNS). The majority of DNS functions are used for DNS resolution in the local domain; however, the UCS BIND integration can also be used for a public name server in principle.

BIND is always available on all domain controller system roles; installation on other system roles is not supported.

The configuration of the name servers to be used by a UCS system is documented in Section 8.2.4.

The following DNS data are differentiated:

- A *forward lookup zone* contains information which is used to resolve DNS names into IP addresses. Each DNS zone has at least one authoritative, primary name server whose information governs the zone. Subordinate servers synchronize themselves with the authoritative server via zone transfers. The entry which defines such a zone is called a *SOA record* in DNS terminology.

- The *MX record* of a forward lookup zone represents important DNS information necessary for e-mail routing. It points to the computer which accepts e-mails for a domain.
- *TXT records* include human-readable text and can include descriptive information about a forward lookup zone.
- A *CNAME record* (also called an alias record) refers to an existing, canonical DNS name. For example, the actual host name of the mail server can be given an alias entry *mailserver*, which is then entered in the mail clients. Any number of CNAME records can be mapped to one canonical name.
- An *A record* (under IPv6 *AAAA record*) assigns an IP address to a DNS name. A records are also known as *Host records* in UCS.
- A *SRV record* (called a service record in UCS) can be used to save information about available system services in the DNS. In UCS, service records are used among other things to make LDAP servers or the master domain controller known domain-wide.
- A *reverse lookup zone* contains information which is used to resolve IP addresses into DNS names. Each DNS zone has at least one authoritative, primary name server whose information governs the zone, subordinate servers synchronize themselves with the authoritative server via zone transfers. The entry which defines such a zone is the *SOA record*.
- A *PTR record (pointer record)* allows resolution of an IP address into a host name. It thus represents the equivalent in a reverse lookup zone of a host record in a forward lookup zone.

## 11.2.1. Configuration of the BIND name server

Feedback

### 11.2.1.1. Configuration of BIND debug output

Feedback

The level of detail of the BIND debug output can be configured via the `dns/debug/level` and `dns/dlz/debug/level` (for the Samba backend, see Section 11.2.1.2) Univention Configuration Registry variables. The possible values are between 0 (no debug tasks) to 11. A complete list of levels can be found at [bind-loglevel].

### 11.2.1.2. Configuration of the data backend

Feedback

In a typical BIND installation on a non-UCS system, the configuration is performed by editing zone files. In UCS, BIND is completely configured via Univention Management Console, which saves its data in the LDAP directory.

BIND can use two different backends for its configuration:

- The *LDAP backend* accesses the data in the LDAP directory. This is the standard backend. The DNS service is split into two in this case: The *BIND proxy* is the primary name server and uses the DNS standard port 53. A second server in the background works on port 7777. If data from the internal DNS zones are edited in the LDAP, the zone file on the second server is updated based on the LDAP information and transmitted to the BIND proxy by means of a zone transfer.
- Samba 4 provides an Active Directory domain. Active Directory is closely connected with DNS, for DNS updates of Windows clients or the localization of NETLOGON shares among other things. If Samba 4 is used, the domain controller in question is switched over to the use of the *Samba backend*. The DNS database is maintained in Samba's internal LDB database, which Samba updates directly. BIND then accesses the Samba DNS data via the DLZ interface.

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. The use of the LDAP backend can thus result in a reduction of the system load on Samba 4 systems.

The backend is configured via the Univention Configuration Registry variable `dns/backend`. The DNS administration is not changed by the backend used and is performed via Univention Management Console in both cases.

#### 11.2.1.3. Configuration of zone transfers

Feedback

In the default setting, the UCS name server allows zone transfers of the DNS data. If the UCS server can be reached from the Internet, a list of all computer names and IP addresses can be requested. The zone transfer can be deactivated when using the OpenLDAP backend by setting the Univention Configuration Registry variable `dns/allow/transfer` to `none`.

### 11.2.2. Administration of DNS data in Univention Management Console

Feedback

DNS files are stored in the `cn=dns,`*`base DN`* container as standard. Forward and reverse lookup zones are stored directly in the container. Additional DNS objects such as pointer records can be stored in the respective zones.

The relative or fully qualified domain name (FQDN) should always be used in the input fields for computers and not the computer's IP address. A FQDN should always end in a full stop to avoid the domain name being added anew.

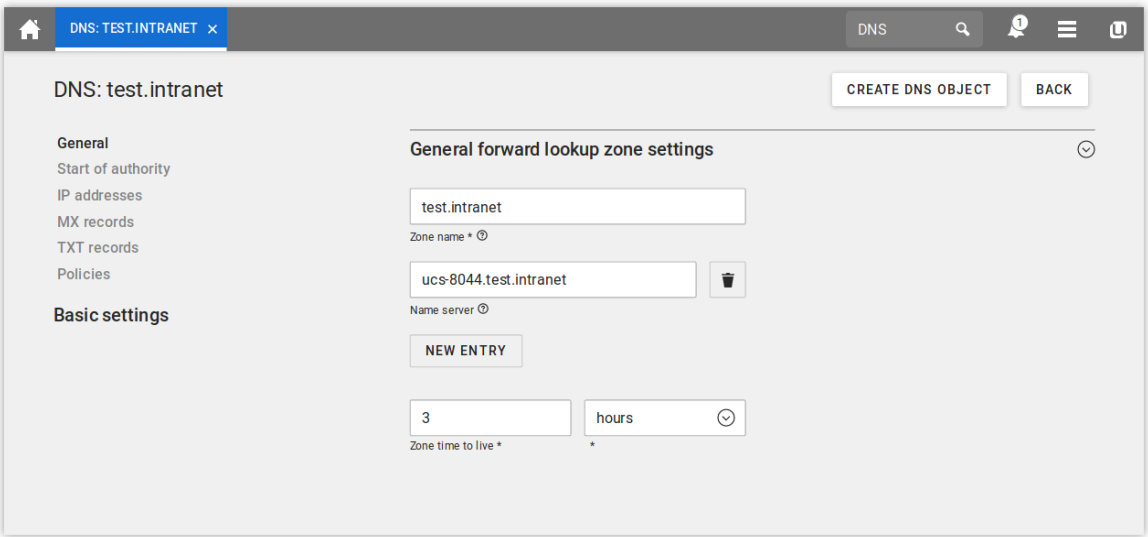
The left column of the UMC module **DNS** includes a list of all the forward and reverse lookup zones. To add an object to a zone - for example an alias record to a forward zone - the corresponding zone must be selected. **Add** is then used to create the object in this zone. To create a new forward or reverse zone, start by selecting **All DNS zones**. Clicking on **Add** then creates a new zone. If an object is created within the zone, the zone is labeled in the UMC dialogues as a *superordinate object*.

#### 11.2.2.1. Forward lookup zone

Feedback

Forward lookup zones contain information which is used to resolve DNS names into IP addresses. They are managed in the UMC module *DNS* (see Section 4.4). To add another forward lookup zone, select **All DNS zones** and **Add -> DNS: Forward lookup zone**.

**Figure 11.2. Configuring a forward lookup zone in UMC**



### Table 11.2. 'General' tab

| Attribute | Description |
|---|---|
| Zone name | This is the complete name of the DNS domain for which the zone will be responsible.<br><br>The domain name must not end in a full stop in zone names! |
| Zone time to live | The time to live specifies how long these files may be cached by other DNS servers. The value is specified in seconds. |
| Name servers | The fully qualified domain name with a full stop at the end of the relative domain name of the responsible name server. The first entry in the line is the primary name server for the zone. |

### Table 11.3. 'Start of authority' tab

| Attribute | Description |
|---|---|
| Contact person | The e-mail address of the person responsible for administrating the zone. |
| Serial number | Other DNS servers use the serial number to recognize whether zone data have changed. The slave name server compares the serial number of its copy with that on the master name server. If the serial number of the slave is lower than that on the master, the slave copies the changed data.<br><br>There are two commonly used patterns for this serial number:<br><br>◦ Start with 1 and increment the serial number with each change<br><br>◦ By including the date the number can be entered in the format ***YYYYMMDDNN***, where *Y* stands for year, *M* for month, *D* for day and *N* for the number of the change of this day.<br><br>If the serial number is not changed manually, it will be increased automatically with every change. |
| Refresh interval | The time span in seconds after which the slave name server checks that its copy of the zone data is up-to-date. |
| Retry interval | The time span in seconds after which the slave name server tries again to check that its copy of the zone data is up-to-date after a failed attempt to update. This time span is usually set to be less than the update interval, but can also be equal. |
| Expiry interval | The time span in seconds after which the copy of the zone data on the slave becomes invalid if it could not be checked to be up-to-date.<br><br>For example, an expiry interval of one week means that the copy of the zone data becomes invalid when all requests to update in one week fail. In this case, it is assumed that the files are too outdated after the expiry interval date to be used further. The slave name server can then no longer answer name resolution requests for this zone. |
| Negative time to live | The negative time to live specifies in seconds how long other servers can cache no-such-domain (NXDOMAIN) answers. This value cannot be set to more than 3 hours, the default value is 3 hours. |

**Table 11.4. 'IP addresses' tab**

| Attribute | Description |
|---|---|
| IP addresses | This input field can be used to specify one or more IP addresses, which are output when the name of the zone is resolved. These IP addresses are queried by Microsoft Windows clients in AD compatible domains. |

**Table 11.5. 'MX records' tab**

| Attribute | Description |
|---|---|
| Priority | A numerical value between 0 and 65535. If several mail servers are available for the MX record, an attempt will be made to engage the server with the lowest priority value first. |
| Mail server | The mail server responsible for this domain as fully qualified domain name with a full stop at the end. Only canonical names and no alias names can be used here. |

**Table 11.6. 'TXT records' tab**

| Attribute | Description |
|---|---|
| TXT record | Descriptive text for this zone. Text records must not contain umlauts or other special characters. |

### 11.2.2.2. CNAME record (Alias records)

Feedback

CNAME records / alias records are managed in the UMC module *DNS* (see Section 4.4). To create another record, the forward lookup zone must be selected in the left column. **Add -> DNS: Alias record** can be used to create a new record.

**Table 11.7. 'General' tab**

| Attribute | Description |
|---|---|
| Alias | The alias name as fully qualified domain name with a full stop at the end or as a relative domain name which should point to the canonical name. |
| Canonical name | The canonical name of the computer that the alias should point to, entered as a fully qualified domain name with a full stop at the end or a relative domain name. |

### 11.2.2.3. A/AAAA records (host records)

Feedback

Host records are managed in the UMC module *DNS* (see Section 4.4). To create another record, the forward lookup zone must be selected in the left column. **Add -> DNS: Host record** can be used to create a new record.

When adding or editing a computer object a host record can be created automatically or edited.

**Table 11.8. 'General' tab**

| Attribute | Description |
|---|---|
| Host name | The FQDN with a full stop at the end or the relative domain name of the name server. |
| IP addresses | The IPv4 and/or IPv6 addresses to which the host record should refer. |

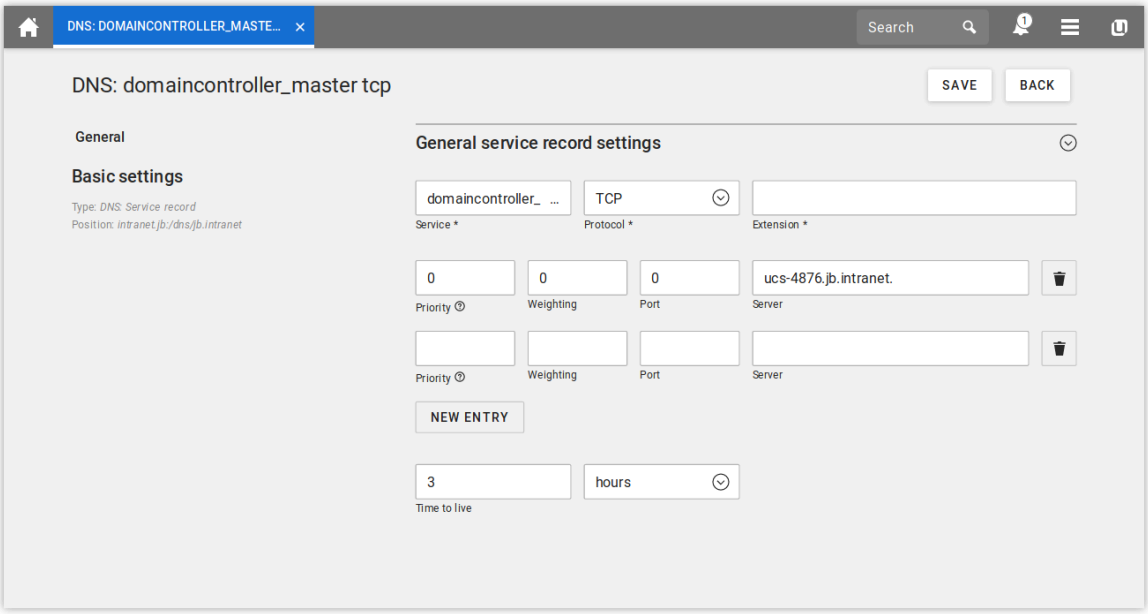
| Attribute | Description |
|---|---|
| Zone time to live | The time to live specifies in seconds how long these files may be cached by other DNS servers. |

### 11.2.2.4. Service records

Feedback

Service records are managed in the UMC module *DNS* (see Section 4.4). To create another record, the forward lookup zone must be selected in the left column. **Add -> DNS: Service record** can be used to create a new record.

**Figure 11.3. Configuring a service record**




A service record must always be assigned to a forward lookup zone and can therefore only be added to a forward lookup zone or a subordinate container.

**Table 11.9. 'General' tab**

| Attribute | Description |
|---|---|
| Service | The name under which the service should be reachable. |
| Protocol | The protocol via which the record can be accessed (`TCP`, `UDP`, `MSDCS` or `SITES`). |
| Extension | This input field can be used to specify additional parameters. |
| Priority | A whole number between 0 and 65535. If more than one server offer the same service, the client will approach the server with the lowest priority value first. |
| Weighting | A whole number between 0 and 65535. The weight function is used for load balancing between servers with the same priority. When more than one server offer the same service and have the same priority the load is distributed across the servers in relation to the weight function.<br><br>Example: `Server1` has a priority of 1 and a weight function of 1, whilst `Server2` also has a priority of 1, but has a weight function of 3. In |

| Attribute | Description |
|---|---|
| | this case, `Server2` will be used three times as often as `Server1`. The load is measured depending on the service, for example, as the number of requests or connection. |
| Port | The port where the service can be reached on the server (valid value from 1 to 65535). |
| Server | The name of the server on which the service will be made available, as a fully qualified domain name with a full stop at the end or a relative domain name.<br><br>Several servers can be entered for each service. |
| Zone time to live | The time to live specifies how long these files may be cached by other DNS servers. |

### 11.2.2.5. Reverse lookup zone

Feedback

A reverse lookup zone is used to resolve IP address into host names. They are managed in the UMC module *DNS*. To add another reverse lookup zone, select **All DNS zones** and **Add -> DNS: Reverse lookup zone**.

**Table 11.10. 'General' tab**

| Attribute | Description |
|---|---|
| Subnet | The IP address of the network for which the reverse lookup zone shall apply. For example, if the network in question consisted of the IP addresses `192.0.2.0` to `192.0.2.255`, **`192.0.2`** should be entered. |
| Zone time to live | The time to live specifies how long these files may be cached by other DNS servers. |

Each DNS zone has at least one authoritative, primary name server whose information governs the zone. Subordinate servers synchronize themselves with the authoritative server via zone transfers. The entry which defines such a zone is called a SOA record in DNS terminology.

**Table 11.11. 'Start of authority' tab**

| Attribute | Description |
|---|---|
| Contact person | The e-mail address of the person responsible for administrating the zone (with a full stop at the end). |
| Name servers | The fully qualified domain name with a full stop at the end or the relative domain name of the primary master name server. |
| Serial number | See the documentation on forward lookup zones in Section 11.2.2.1. |
| Refresh interval | See the documentation on forward lookup zones in Section 11.2.2.1. |
| Retry interval | See the documentation on forward lookup zones in Section 11.2.2.1. |
| Expiry interval | See the documentation on forward lookup zones in Section 11.2.2.1. |
| Minimum time to live | See the documentation on forward lookup zones in Section 11.2.2.1. |

### 11.2.2.6. Pointer record

Feedback

Pointer records are managed in the UMC module *DNS* (see Section 4.4). To create another record, the reverse lookup zone must be selected in the left column. **Add -> DNS: Pointer record** can be used to create a new record.

**Table 11.12. 'General' tab**

| Attribute | Description |
|---|---|
| Address | The last octet of the computer's IP address (depends on network prefix, see example below). |
| Pointer | The computer's fully qualified domain name with a full stop at the end.<br><br>In a network with a 24-bit network prefix (subnet mask `255.255.255.0`) a pointer should be created for the `client001` computer with the IP address `192.0.2.101`. **`101`** must then be entered in the **Address field** and **`client001.company.com.`** in **Pointer**.<br><br>Example:<br><br>For a network with a 16-bit network prefix (subnet mask `255.255.0.0`) the last two octets should be entered in reverse order for this computer (here **`101.1`**). **`client001.company.com.`** also needs to be entered in the **Pointer** field here. |

# 11.3. IP assignment via DHCP

Feedback

## 11.3.1. Introduction

Feedback

The *Dynamic Host Configuration Protocol* (DHCP) assigns computers an IP address, the subnet mask and further settings for the gateway or NetBIOS server as necessary. The IP address can be set fixed or dynamic.

The use of DHCP allows central assignment and control of IP addresses via the LDAP directory without performing manual configuration on the individual computer systems.

The DHCP integration in UCS only supports IPv4.

In a *DHCP service*, DHCP servers are grouped in a shared LDAP configuration. Global configuration parameters are entered in the DHCP service; specific parameters in the subordinate objects.

A DHCP server can be installed from the Univention App Center with the application *DHCP server*. Alternatively, the software package ***univention-dhcp*** can be installed. Additional information can be found in Section 5.6.

Every DHCP assigns IP addresses via DHCP. In the default setting, only static IP addresses are assigned to computer objects registered in the UCS LDAP.

If only fixed IP addresses are assigned, as many DHCP servers as required may be used in a DHCP service. All the DHCP servers procure identical data from the LDAP and offer the DHCP clients the data multiple times. DHCP clients then accept the first answer and ignore the rest.

If dynamic IP addresses are also assigned, the DHCP failover mechanism must be employed and a maximum of two DHCP servers can be used per subnet.

A *DHCP host* entry is used to make the DHCP service aware of a computer. A DHCP host object is required for computers attempting to retrieve a fixed IP address over DHCP. DHCP computer objects do not normally need to be created manually, because they are created automatically when a DHCP service is assigned to a computer object with a fixed IP address.

A *DHCP subnet* entry is required for every subnet, irrespective of whether dynamic IP addresses are to be assigned from this subnet.

Configuration parameters can be assigned to the different IP ranges by creating *DHCP pools* within subnets. In this way unknown computers can be allowed in one IP range and excluded from another IP range. DHCP pools can only be created below DHCP subnet objects.

If several IP subnets are used in a physical Ethernet network, this should be entered as a *DHCP shared subnet* below a **DHCP shared network**. **DHCP shared subnet** objects can only be created below **DHCP shared network** objects.

Values which are set on a DHCP configuration level always apply for this level and all subordinate levels, unless other values are specified there. Similar to policies, the value which is closest to the object always applies.

## 11.3.2. Composition of the DHCP configuration via DHCP LDAP objects

Feedback

The left column of the UMC module **DHCP** includes a list of all the DHCP services. To add an object to a DHCP service - for example in an additional subnet - the corresponding service must be selected. **Add** is then used to create the object in this service. To create a new DHCP service, start by selecting **All DHCP services**. Clicking on **Add** then creates a new service. If an object is saved within a service, the service is labeled in UMC dialogues as a *superordinate object*.

### 11.3.2.1. Administration of DHCP services

Feedback

DHCP services are managed in the UMC module *DHCP* (see Section 4.4). To create a new DHCP service, **All DHCP services** needs to be selected in the left column of the UMC module. Clicking on **Add** then creates a new service.

A DHCP server can only serve one DHCP service; to use another DHCP service, a separate DHCP server must be set up (see Section 11.3.2.2).

The following parameters are often set on the DHCP service object which then apply to all the computers which are served by this DHCP service (unless other values are entered in lower levels):

- **Domain name** and **Domain name servers** under **Policy: DHCP DNS**
- **NetBIOS name servers** under **Policy: DHCP NetBIOS**

A description of this and the other DHCP policies can be found at Section 11.3.3.

**Table 11.13. 'General' tab**

| Attribute | Description |
|---|---|
| Service name | An unambiguous name for the DHCP service must be entered in this input field, e.g., `company.example`. |

### 11.3.2.2. Administration of DHCP server entries

Feedback

Each server which should offer the DHCP service requires a *DHCP server* entry in the LDAP directory. The entry does not normally need to be created manually, instead it is created by the join script of the ***univention-dhcp*** package. However, to create another record manually, a DHCP service must be selected in the left column of the UMC module *DHCP*. **Add -> DHCP Server** can then be used to register a new server.

**Table 11.14. 'General' tab**

| Attribute | Description |
|---|---|
| Server name | The computer name that the DHCP service should offer is entered in this input field, e.g., `ucs-master`.<br><br>A server can only ever provide a single DHCP service and therefore cannot be entered in more than one DHCP service at the same time. |

### 11.3.2.3. Administration of DHCP subnets

Feedback

DHCP subnets are managed in the UMC module *DHCP* (see Section 4.4). To create another subnet, a DHCP service must be selected in the left column. **Add -> DHCP: Subnet** can be used to create a new subnet.

A DHCP subnet entry is required for every subnet from which dynamic or fixed IP addresses are to be assigned. It is only necessary to enter IP address ranges if IP addresses are to be assigned dynamically.

If *DHCP shared subnet* objects are to be used, the corresponding subnets should be created below the *DHCP shared subnet* container created for this purpose (see Section 11.3.2.6).

**Table 11.15. 'General' tab**

| Attribute | Description |
|---|---|
| Subnet address | The IP address of the subnet must be entered in dot-decimal form in this input field, e.g., `192.0.2.0`. |
| Net mask | The network mask can be entered in this input field as the network prefix or in dot-decimal form. If the network mask is entered in dot-decimal form it will be subsequently be converted into the corresponding network prefix and later also shown so. |
| Dynamic address assignment | Here one can set up individual or multiple IP address ranges for dynamic assignment. The range stretches from the **First address** to the **Last address** in dot-decimal form.<br><br>**Caution**<br><br>Dynamic IP ranges for a subnet should always either be specified exclusively in the subnet entry or exclusively in one or more special pool entries. The types of IP range entries within a subnet must not be mixed! If different IP ranges with different configurations are be set up in one subnet, pool entries must be created for this purpose. |

At this level, the gateway for all computers in a subnet is often set using the **Policy: DHCP Routing** tab (unless other entries are performed at lower levels).

### 11.3.2.4. Administration of DHCP pools

Feedback

DHCP pools can only be managed via the UMC module *LDAP directory*. To do so, one must always be in a DHCP subnet object - a DHCP pool object must always be created below a DHCP subnet object - and a **DHCP: Pool** object added with **Add**.

If DHCP pools are created in a subnet, no IP address range should be defined in the subnet entry. These should only be specified via the pool entries.

**Table 11.16. 'General' tab**

| Attribute | Description |
|---|---|
| Name | An unambiguous name for the DHCP pool must be entered in this input field, e.g., `testnet.compaby.example`. |
| Dynamic range | Here you can enter the IP addresses in dot-decimal form that are to be dynamically assigned. |

**Table 11.17. 'Advanced settings' tab**

| Attribute | Description |
|---|---|
| Failover peer | The name of a failover configuration, which must to be configured manually in file `/etc/dhcp/local.conf`. Further information can be found at [dhcp-failover]. |
| Allow known clients | A computer is identified by its MAC address. If this input field is set to **allow** or unset, a computer with a matching DHCP host entry (see Section 11.3.2.5) is eligible to receive an IP address from this pool. If set to **deny**, the computer doesn't receive an IP address from the pool. |
| Allow unknown clients | A computer is identified by its MAC address. If this input field is set to **allow** or unset, a computer without a matching DHCP host entry (see Section 11.3.2.5) is eligible to receive an IP address from this pool. If set to **deny**, the computer doesn't receive an IP address from the pool. |
| Allow dynamic BOOTP clients | *BOOTP* is the predecessor of the DHCP protocol. It has no mechanism to renew leases and by default assigns leases infinitely, which can deplete the pool. If this options is set to **allow** clients can retrieve an IP address from this pool using BOOTP. |
| All clients | If this option is set to **deny** the pool is disabled globally. This is only useful in exceptional scenarios. |

### 11.3.2.5. Registration of computers with DHCP computer objects

Feedback

A *DHCP host* entry is used to register the respective computer in the DHCP service. Computers can be handled depending on their registration status. Known computers may get fixed and dynamic IP addresses from the DHCP service; unknown computers only get dynamic IP addresses.

DHCP computer entries are usually created automatically when a computer is added via the computer management. Below the DHCP service object you have the possibility of adding DHCP computer entries or editing existing entries manually, irrespective of whether they were created manually or automatically.

DHCP host objects are managed in the UMC module *DHCP* (see Section 4.4). To register a host in the DHCP manually, a DHCP service must be selected in the left column of the module. **Add -> DHCP: Host** can be used to register a host.

**Table 11.18. 'General' tab**

| Attribute | Description |
|---|---|
| Host name | A name for the computer is entered in this input field (which usually also has an entry in the computer management). It is recommended to enter the same name and the same MAC address for the computer in both entries to facilitate assignment. |
| Type | The type of network used can be selected in this select list. **Ethernet** almost always needs to be selected here. |

| Attribute | Description |
|---|---|
| Address | The MAC address of the network card needs to be entered here, e.g., `2e:44:56:3f:12:32` or `2e-44-56-3f-12-32`. |
| Fixed IP addresses | One or more fixed IP addresses can be assigned to the computer here. In addition to an IP address, a fully qualified domain names can also be entered, which is resolved into one or more IP addresses by the DHCP server. |

### 11.3.2.6. Management of DHCP shared networks / DHCP shared subnets

Feedback

*DHCP shared network* objects accept subnets which use a common physical network.

DHCP shared network objects are managed in the UMC module *DHCP* (see Section 4.4). To create a shared network, a DHCP service must be selected in the left column of the module. **Add -> DHCP: Shared Network** can be used to register a network.

#### Caution

A shared network must contain at least one shared subnet object. Otherwise the DHCP service will terminate itself and cannot be restarted until the configuration is fixed.

**Table 11.19. 'General' tab**

| Attribute | Description |
|---|---|
| Shared network name | A name for the shared network must be entered in this input field. |

Subnets are declared as a *DHCP shared subnet* when they use the same, common physical network. All subnets which use the same network must be stored below the same shared network container. A separate *DHCP shared subnet* object must be created for each subnet.

DHCP shared subnet objects can only be managed via the UMC module *LDAP directory*. To do so, one must always be in a DHCP shared network object - a DHCP shared subnet object must always be created below a DHCP shared network object - and a **DHCP shared subnet** object added with **Add**.

## 11.3.3. Configuration of clients via DHCP policies

Feedback

#### Note

Many of the settings for DHCP are configured via policies. They are also applied to DHCP computer objects if a policy is linked to the LDAP base or one of the other intermediate containers. As the settings for DHCP computer objects have the highest priority, other settings for subnetwork and service objects are ignored.

For this reason, DHCP policies should be linked directly to the DHCP network objects (e.g., the DHCP subnetworks). [1]

#### Tip

When using the command line `udm dhcp/host list` (see also Section 4.10.2.7), it is possible to use the option `--policies 0` to display the effective settings.

[1] Alternatively, the LDAP class `univentionDhcpHost` can be added in the advanced settings of the policies under **Object** and then **Excluded object classes**. Such policies are then no longer applied to the DHCP computer objects, with the result that the settings from the DHCP subnetwork and service are used.

### 11.3.3.1. Setting the gateway

Feedback

The default gateway can be specified via DHCP with a *DHCP routing* policy, which is managed in the UMC module **Policies** (see Section 4.6)

**Table 11.20. 'General' tab**

| Attribute | Description |
|---|---|
| Routers | The names or IP addresses of the routers are to be entered here. It must be verified that the DHCP server can resolve these names in IP addresses. The routers are contacted by the client in the order in which they stand in the selection list. |

### 11.3.3.2. Setting the DNS servers

Feedback

The name servers to be used by a client can be specified via DHCP with a *DHCP DNS* policy, which is managed in the UMC module **Policies** (see Section 4.6)

**Table 11.21. 'General' tab**

| Attribute | Description |
|---|---|
| Domain name | The name of the domain, which the client automatically appends on computer names that it sends to the DNS server for resolution and which are not FQDNs. Usually this is the name of the domain to which the client belongs. |
| Domain name servers | Here IP addresses or fully qualified domain names (FQDNs) of DNS servers can be added. When using FQDNs, it must be verified that the DHCP server can resolve the names in IP addresses. The DNS servers are contacted by the clients according to the order specified here. |

### 11.3.3.3. Setting the WINS server

Feedback

The WINS server to be used can be specified via DHCP with a *DHCP NetBIOS* policy, which is managed in the UMC module **Policies** (see Section 4.6)

**Table 11.22. 'General' tab**

| Attribute | Description |
|---|---|
| NetBIOS name servers | The names or IP addresses of the NetBIOS name servers (also known as WINS servers) should be entered here. It must be verified that the DHCP server can resolve these names in IP addresses. The servers entered are contacted by the client in the order in which they stand in the selection list. |
| NetBIOS scope | The NetBIOS over TCP/IP scope for the client according to the specification in RFC 1001[1] and RFC 1002[1]. Attention must be paid to upper-case and lowercase when entering the NetBIOS scope. |
| NetBIOS node type | This field sets the node type of the client. Possible values are:<br>◦ **1 B-node** (Broadcast: no WINS) |

[1] http://tools.ietf.org/html/rfc1001
[1] http://tools.ietf.org/html/rfc1002

| Attribute | Description |
|---|---|
| | ◦ **2 P-node** (Peer: only WINS)<br>◦ **4 M-node** (Mixed: first Broadcast, then WINS)<br>◦ **8 H-node** (Hybrid: first WINS, then Broadcast) |

### 11.3.3.4. Configuration of the DHCP lease

Feedback

The validity of an assigned IP address - a so-called DHCP lease - can be specified with a *DHCP lease time* policy, which is managed in the UMC module **Policies** (see Section 4.6)

**Table 11.23. 'General' tab**

| Attribute | Description |
|---|---|
| Default lease time | If the client does not request a specific lease time, the standard lease time is assigned. If this input field is left empty, the DHCP server's default value is used. |
| Maximum lease time | The maximum lease time specifies the longest period of time for which a lease can be granted. If this input field is left empty, the DHCP server's default value is used. |
| Minimum lease time | The minimum lease time specifies the shortest period of time for which a lease can be granted. If this input field is left empty, the DHCP server's default value is used. |

### 11.3.3.5. Configuration of boot server/PXE settings

Feedback

A *DHCP Boot* policy is used to assign computers configuration parameters for booting via BOOTP/PXE. They are managed in the UMC module **Policies** (see Section 4.6)

**Table 11.24. 'Boot' tab**

| Attribute | Description |
|---|---|
| Boot server | The IP address or the FQDN of the PXE boot server from which the client should load the boot file is entered in the input field. If no value is entered in this input field, the client boots from the DHCP server from which it retrieves its IP address. |
| Boot filename | The path to the boot file is entered here. The path must be entered relative to the base directory of the TFTP service (`/var/lib/univention-client-boot/`). |

### 11.3.3.6. Further DHCP policies

Feedback

There are some further DHCP policies available, but they are only required in special cases.

◦ *DHCP Dynamic DNS* allows the configuration of dynamic DNS updates. These cannot yet be performed with a LDAP-based DNS service as provided out-of-the-box by UCS.

◦ *DHCP Allow/Deny* allows the configuration of different DHCP options, which control what clients are allowed to do. The are only useful in exceptional cases.

◦ *DHCP statements* allows the configuration of different options, which are only required in exceptional cases.

## 11.4. Packet filter with Univention Firewall

Feedback

Univention Firewall integrates a packet filter based on ***iptables*** in Univention Corporate Server.

It permits targeted filtering of undesired services and the protection of computers during installations. Furthermore it provides the basis for complex scenarios such as firewalls and application level gateways. Univention Firewall is included in all UCS installations as standard.

In the default setting, all incoming ports are blocked. Every UCS package provides rules, which free up the ports required by the package again.

The configuration is primarily performed via Univention Configuration Registry variables. The definition of this type of packet filter rules is documented in [developer-reference].

In addition, the configuration scripts in the `/etc/security/packetfilter.d/` directory are listed in alphabetic order. The names of all scripts begin with two digits, which makes it easy to create a numbered order. The scripts must be marked as executable.

After changing the packet filter settings, the ***univention-firewall*** service has to be restarted.

Univention Firewall can be deactivated by setting the Univention Configuration Registry variable `security/packetfilter/disabled` to `true`

## 11.5. Web proxy for caching and policy management / virus scan

Feedback

The UCS proxy integration allows the use of a web cache for improving the performance and controlling data traffic. It is based on the tried-and-tested proxy server Squid and supports the protocols HTTP, FTP and HTTPS.

A proxy server receives requests about Internet contents and verifies whether these contents are already available in a local cache. If this is the case, the requested data are provided from the local cache. If the data are not available, these contents are called up from the respective web server and inserted in the local cache. This can be used to reduce the answering times for the users and the transfer volume via the Internet access.

The software DansGuardian can be installed as an additional component with the package ***univention-dansguardian***. This makes it possible to check and filter the Internet contents prior to delivery to the user in order to scan files for viruses or prevent access to undesirable content.

Further documentation on proxy services - such as the cascading of proxy servers, transparent proxies and the integration of a virus scan engine - are documented in [ext-doc-net].

### 11.5.1. Installation

Feedback

Squid can be installed from the Univention App Center with the application *Web proxy / web cache (Squid)*. Alternatively, the software package ***univention-squid*** can be installed. Additional information can be found in Section 5.6.

The service is configured with standard settings sufficient for operation so that it can be used immediately. It is possible to configure the port on which the service is accessible to suit your preferences (see Section 11.5.5.1); port 3128 is set as default.

If changes are made to the configuration, Squid must be restarted. This can be performed either via Univention Management Console or the command line:

```
/etc/init.d/squid restart
```

In addition to the configuration possibilities via Univention Configuration Registry described in this document, it is also possible to set additional Squid configuration options in the `/etc/squid/local.conf`.

*DansGuardian* can be installed via the package ***univention-dansguardian***, see Section 11.5.7.

## 11.5.2. Caching of web content

Feedback

Squid is a caching proxy, i.e., previously viewed contents can be provided from a cache without being reloaded from the respective web server. This reduces the incoming traffic via the Internet connection and can result in quicker responses of HTTP requests.

However, this caching function is not necessary for some environments or, in the case of cascaded proxies, it should not be activated for all of them. For these scenarios, the caching function of the Squid can be deactivated with the Univention Configuration Registry variable `squid/cache` by setting this to `no`. Squid must then be restarted.

## 11.5.3. Logging proxy accesses

Feedback

All accesses performed via the proxy server are stored in the logfile `/var/log/squid/access.log`. It can be used to follow which websites have been accessed by the users.

When DansGuardian is used, all accesses are documented in `/var/log/dansguardian/access.log`.

## 11.5.4. Restriction of access to permitted networks

Feedback

As standard, the proxy server can only be accessed from local networks. If, for example, a network interface with the address `192.0.2.10` and the network mask `255.255.255.0` is available on the computer on which Squid is installed, only computers from the network `192.0.2.0/24` can access the proxy server. Additional networks can be specified via the Univention Configuration Registry variable `squid/allowfrom`. When doing so, the CIDR notation must be used; several networks should be separated by blank spaces.

Example:

```
univention-config-registry set squid/allowfrom="192.0.2.0/24
 192.0.2.0/24"
```

Once Squid has been restarted, access is now permitted from the networks `192.0.2.0/24` and `192.0.2.0/24`. If configured to `all`, proxy access in granted from all networks.

If Squid is used together with DansGuardian, i.e., the virus or web content filter is activated, Squid cannot verify the access as the connections are performed via DansGuardian. In this case, the access can be restricted via DansGuardian.

## 11.5.5. Configuration of the ports used

Feedback

### 11.5.5.1. Access port

Feedback

As standard, the web proxy can be accessed via port 3128. If another port is required, this can be configured via the Univention Configuration Registry variable `squid/httpport`. If Univention Firewall is used, the packet filter configuration must also be adjusted.

When using the content and virus scanner (see Section 11.5.7) DansGuardian is accessible at the configured port instead of Squid. Squid then occupies the next-highest port. This must be borne in mind if there are other applications which are supposed to offer services via this port.

#### 11.5.5.2. Permitted ports

Feedback

In the standard configuration, Squid only forwards client requests intended for the network ports 80 (HTTP), 443 (HTTPS) or 21 (FTP). The list of permitted ports can be changed via the Univention Configuration Registry variable; several entries should be separated by blank spaces.

Example:

```
univention-config-registry set squid/webports="80 443"
```

With this setting, access is only allowed to ports 80 and 443 (HTTP and HTTPS).

### 11.5.6. User authentication on the proxy

Feedback

It is sometimes necessary to restrict web access to certain users. Squid allows user-specific access regulation via group memberships. To allow verification of group membership, it is necessary for the user to authenticate on the proxy server.

> **Caution**
>
> To prevent unauthorized users from opening websites nonetheless, additional measures are required to prevent these users from bypassing the proxy server and accessing the Internet. This can be done, for example, by limiting all HTTP traffic through a firewall.

The proxy authentication (and as a result the possible verification of the group memberships) must firstly be enabled. There are three possible mechanisms for this:

- Direct authentication against the LDAP server. This is done by setting the Univention Configuration Registry variable `squid/basicauth` to `yes` and restarting Squid.
- Authentication is performed via the NTLM interface. Users logged in on a Windows client then do not need to authenticate themselves again when accessing the proxy. NTLM authentication is enabled by setting the Univention Configuration Registry variable `squid/ntlmauth` to `yes` and restarting Squid.
- Authentication is performed via Kerberos. Users logged in on a Windows client which is a member of a Samba 4 domain authenticate themselves on the proxy with the ticket that they received when they logged in to the domain. The ***univention-squid-kerberos*** package must be installed on every proxy server for it to be possible to enable Kerberos authentication. Then the Univention Configuration Registry variable `squid/krb5auth` must be set to `yes` and Squid restarted.

If NTLM is used an NTLM authentication is performed for every HTTP query as standard. If for example the website `https://www.univention.de/` is opened, the subpages and images are loaded in addition to the actual HTML page. The NTLM authentication can be cached per domain: If the Univention Configuration Registry variable `squid/ntlmauth/keepalive` is set to `yes`, no further NTLM authentication is performed for subsequent HTML queries in the same domain. In case of problems with local user accounts it may help to set this variable to `no`.

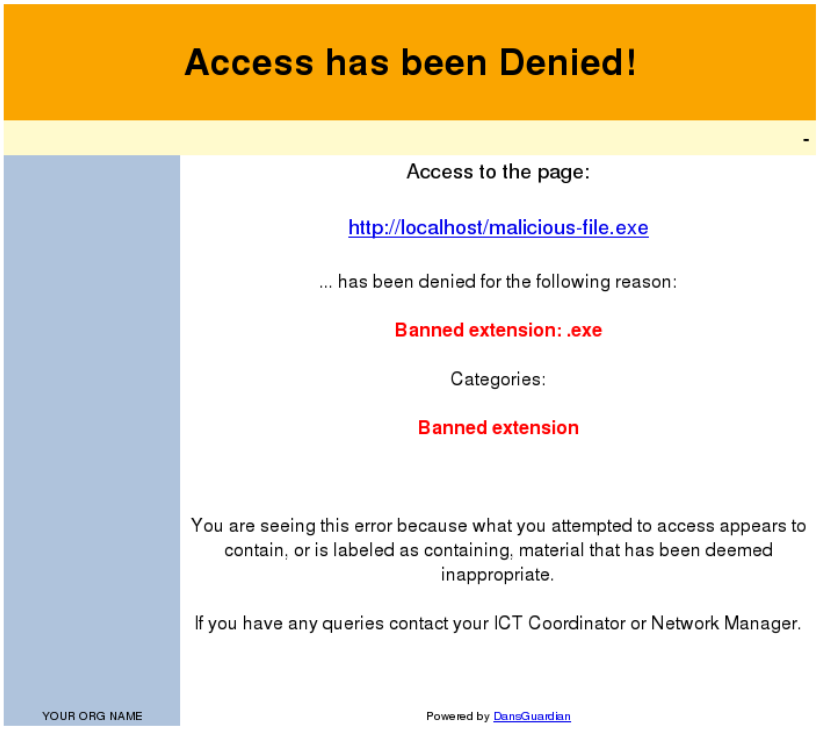
In the standard setting all users can access the proxy. The Univention Configuration Registry variable `squid/auth/allowed_groups` can be used to limit the proxy access to one or several groups. If several groups are specified, they must be separated by a semicolon.

### 11.5.7. Filtering/policy enforcement of web content with DansGuardian

Feedback

DansGuardian accepts website requests from the network and checks whether access by the sender of the request is permitted. If so, the request is forwarded to the proxy server, Squid. For example, DansGuardian allows the blocking of individual file types and suffixes or access to websites or domains.

**Figure 11.4. DansGuardian blocking a web site**




It is also possible to scan requested files for viruses. In the default setting, the free virus scanner ClamAV is used. The setup is documented in the extended network management documentation [ext-doc-net].

### Caution

Direct access to the proxy server Squid is restricted to access from the local host. Users working on the system on which Squid and DansGuardian are installed have the possibility of bypassing the filter functions by accessing Squid directly. The web proxy and DansGuardian should thus only be installed on dedicated systems which users cannot log in to.

Following the installation of ***univention-dansguardian***, the virus scanner and the filter for web contents are activated.

The filtering of web content and the virus scanner can be activated separately. In order to deactivate the content filter, the Univention Configuration Registry variable `squid/contentscan` must be set to `no` and Squid restarted. To disable the virus scanner, the Univention Configuration Registry variable `squid/virusscan` must be set to `no`. If neither of the two variables is set to `yes`, DansGuardian is not used. After changes to the variables Squid and - if available - DansGuardian must be restarted.

## 11.5.8. Definition of content filters for DansGuardian

Feedback

Web content can be filtered based on file suffixes, MIME types, websites and individual URLs. It is possible to exempt individual computers or users from the filtering.

The filter function can be configured via the following Univention Configuration Registry variables. Where several values are to be added, these must each be separated by blank spaces. The filtering is performed on the basis of group memberships, i.e., different rules can be defined per group and as such it is possible to realize different rights when accessing the Internet. Which groups are taken into account by DansGuardian can be defined in the Univention Configuration Registry variable `dansguardian/groups`.

It must be noted that the first group in the list plays a special role. All users which cannot be assigned to one of the specified groups are assigned to this one, i.e., the defined filter rules apply. This group is generally assigned the lowest rights.

For group changes to take effect, DansGuardian needs to be restarted. This can be done either in the UMC module **System services** or on the command line using the command

```
/etc/init.d/dansguardian restart
```

For changes to filter rules, it is sufficient to reload the configuration using the following command:

```
dansguardian -g
```

The Univention Configuration Registry variables for the definition of the filter rules contain the group names replaced in the following list by *group*.

### Table 11.25. UCR variables for filter rules

| UCR variable | Description |
|---|---|
| dansguardian/groups/*group*/banned/extensions | Files with the specified file suffixes may not be downloaded. The suffix point must always be specified. If this variable is left blank, standard values are used. To allow all file suffixes, the variable must be set to ' ' (string with a blank space). Example: **'.doc .xls .exe'**. |
| dansguardian/groups/*group*/banned/mimetypes | Files with the specified MIME types may not be downloaded. The MIME type is specified by the delivering web server (or an application running on it). Normally, the MIME types corresponding to the file suffixes outlined above are specified. If this variable is left blank, standard values are used. To allow all MIME types, the variable must be set to ' ' (string with a blank space). Example: *audio/mpeg application/zip* |
| dansguardian/groups/*group*/banned/sites | This can be used to block complete web sites. Example: illegal-example-website.com |
| dansguardian/*group*/banned/urls | In contrast to the previous parameter, this can be used to block only specific URLs of websites. |
| dansguardian/*group*/exception/urls | The access to the URLs specified here is not filtered by DansGuardian. |
| dansguardian/*group*/exception/sites | The access to the web sites specified here is not filtered by DansGuardian. |
| dansguardian/bannedipaddresses | This variable makes it possible to exclude individual clients (based on the IP address) from accessing the proxy server |
| dansguardian/exceptionipaddresses | This can be used to disable all filter rules for individual computers with the result that all the files can be downloaded from the proxy server from this computer. This can be useful if, for example, an administration computer should be used to download files for other users. |

## Caution

The definition of an exception rule for content filters using dansguardian/*group*/exception/* also exempts the content from virus scanning!

# 11.6. RADIUS

Feedback

The ***RADIUS*** app increases the security for UCS managed IT infrastructures by controlling the access to the wireless network for users, groups and endpoint devices via RADIUS protocol[2]. The configuration is done via black and white lists and directly at user, group and endpoint device objects in the UCS management system. Registered users are authenticated with their usual domain credentials, which, among others, also allows bring your own device concepts.

## 11.6.1. Installation

Feedback

***RADIUS*** is available through the App Center (see Section 5.3) and can be installed using the corresponding Univention Management Console module App Center. It can be installed on multiple machines. After the installation it runs a FreeRADIUS[3] server. Authenticators (e.g. access points) can contact via RADIUS to check network access requests.

The RADIUS app can also be installed on UCS@school systems. In this case, the network access can be given to users or groups regardless of the Internet rule or computer room settings.

## 11.6.2. Configuration

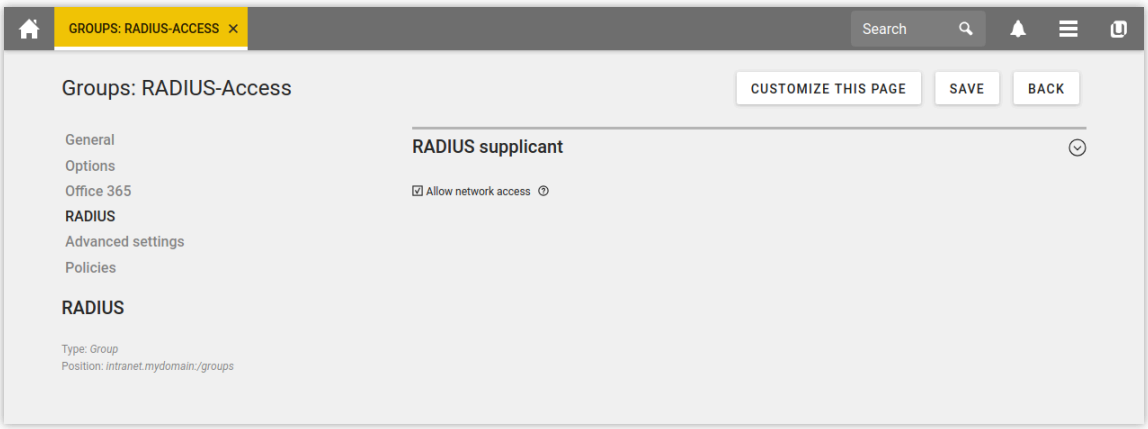
Feedback

### 11.6.2.1. Allowed users

Feedback

By default no user is allowed to access the network. Enabling the checkbox for **network access** on the **RADIUS** tab, gives the user access to the network. The checkbox can also be set on groups, which allows all users in this group access.

**Figure 11.5. Example for a group allowing network access to its users**




### 11.6.2.2. MAC filtering

Feedback

By default access to the network is allowed for every device (assuming the used user name has access). It can be restricted to only allow specific devices. This can be enabled by setting the Univention Configuration Registry variable `radius/mac/whitelisting` to `true`. When enabled, the device used to access the network is looked up via the LDAP attribute *macAddress* and the resulting computer object must have network access granted (either directly or via one of its groups), too.

[2] https://en.wikipedia.org/wiki/RADIUS
[3] http://freeradius.org/

### 11.6.2.3. Access point administration

All access points must be known to the RADIUS server. An access point can either be configured in the file `/etc/freeradius/3.0/clients.conf` or through the Univention Management Console. For each access point a random shared secret should be created (e.g. by using the command `makepasswd`). The `shortname` can be chosen at will.

Example entry for an access point:

```
client AP01 {
    secret = a9RPAeVG
    ipaddr = 192.0.2.101
}
```

To configure an access point using the Univention Management Console a computer object must exists for it. This object needs the *RADIUS-Authenticator* option to be set (Figure 11.6). An *IP managed client* is a good choice as a computer object for access points. The RADIUS settings can be edited on the **RADIUS** tab of the object (Figure 11.7). At least the IP address and the shared secret must be configured. The virtual server and NAS type options usually do not need to be changed.

Access points that are configured via the UMC are available to all RADIUS server in the domain. To achieve this, the Univention Directory Listener will write them into the file `/etc/freeradius/3.0/clients.univention.conf` and restart the RADIUS server. In order to merge multiple changes in one restart, this happens with a slight delay (around 15 seconds). New access points can only access the RADIUS server after this restart.

**Figure 11.6. RADIUS option**

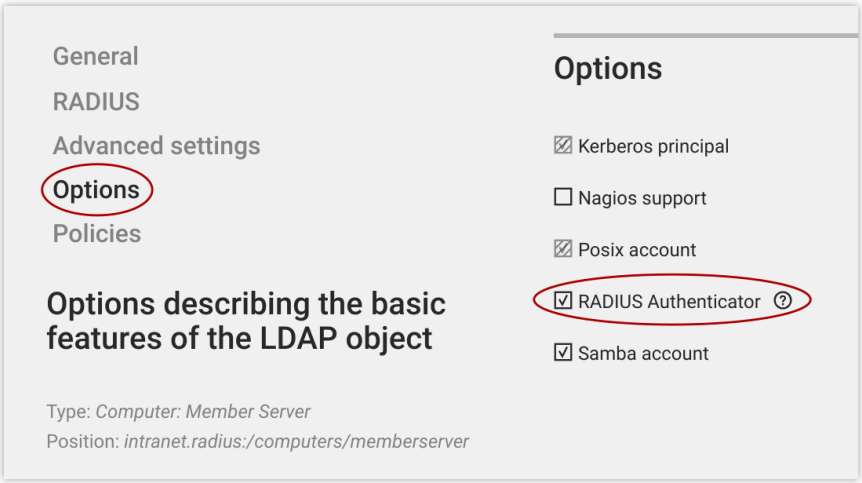
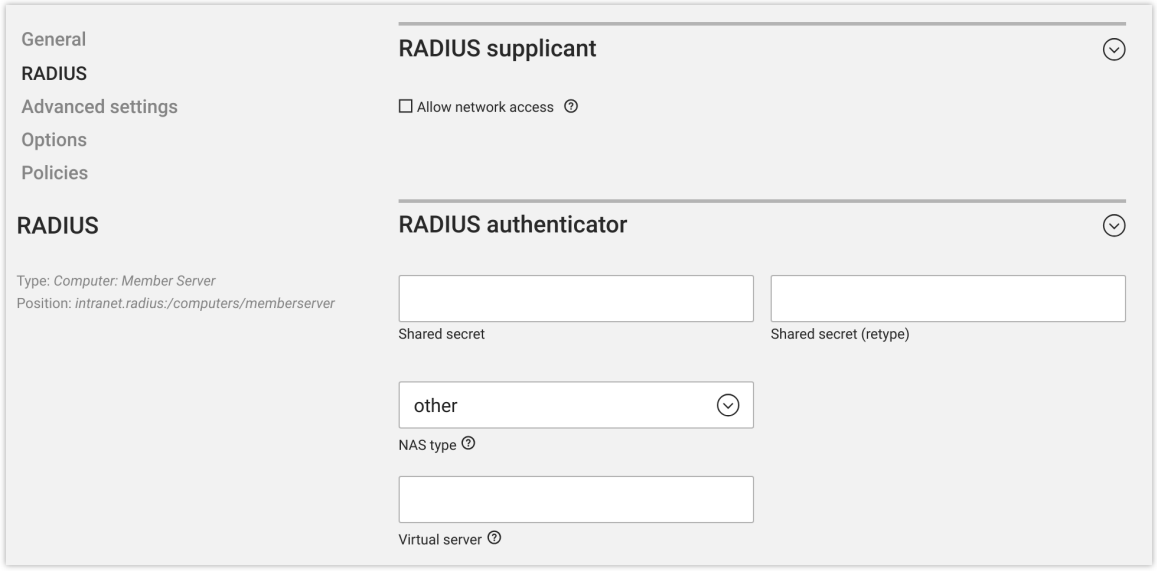


**Figure 11.7. RADIUS authenticator options**




### 11.6.2.4. Access point and client configuration

Feedback

The access points must then be configured to use 802.1x ("WPA Enterprise") authentication. And the "RADIUS server" address should be set to the address of the server, where the RADIUS app is installed. The password must be set to the `secret` from the `clients.conf` entry for that access point.

WiFi clients have to be configured to use *WPA* with *PEAP* and *MSCHAPv2* for authentication.

## 11.6.3. Debugging

Feedback

The ***RADIUS*** app has a log file under `/var/log/univention/radius_ntlm_auth.log`. The log verbosity can the controlled via the Univention Configuration Registry variable `freeradius/auth/helper/ntlm/debug`. The ***FreeRADIUS server*** uses the log file: `/var/log/freeradius/radius.log`.

The tool ***univention-radius-check-access*** can be used to evaluate the current access policy for a given user and/or station ID (MAC address). It can be executed as root on the server where ***univention-radius*** its installed:

```
root@master211:~# univention-radius-check-access --username=stefan
DENY 'uid=stefan,cn=users,dc=ucs,dc=local'
'uid=stefan,cn=users,dc=ucs,dc=local'
-> DENY 'cn=Domain Users,cn=groups,dc=ucs,dc=local'
-> 'cn=Domain Users,cn=groups,dc=ucs,dc=local'
-> -> DENY 'cn=Users,cn=Builtin,dc=ucs,dc=local'
-> -> 'cn=Users,cn=Builtin,dc=ucs,dc=local'
Thus access is DENIED.
```

```
root@master211:~# univention-radius-check-access --username=janek
DENY 'uid=janek,cn=users,dc=ucs,dc=local'
'uid=janek,cn=users,dc=ucs,dc=local'
-> DENY 'cn=Domain Users,cn=groups,dc=ucs,dc=local'
-> ALLOW 'cn=Network Access,cn=groups,dc=ucs,dc=local'
-> 'cn=Domain Users,cn=groups,dc=ucs,dc=local'
-> -> DENY 'cn=Users,cn=Builtin,dc=ucs,dc=local'
```

```
-> -> 'cn=Users,cn=Builtin,dc=ucs,dc=local'
-> 'cn=Network Access,cn=groups,dc=ucs,dc=local'
Thus access is ALLOWED.
root@master211:~#
```

It prints a detailed explanation and sets the exit code depending on the result of the access check (0 for *access granted*, 1 for *access denied*).

# Chapter 12. File share management

12.1. Access rights to data in shares ..... 227
12.2. Management of shares in UMC ..... 228
12.3. Support for MSDFS ..... 235
12.4. Configuration of file system quota ..... 235
12.4.1. Activating filesystem quota ..... 236
12.4.2. Configuring filesystem quota ..... 236
12.4.3. Evaluation of quota during login ..... 237
12.4.4. Querying the quota status by administrators or users ..... 237

UCS supports the central management of directory shares. A share registered in Univention Management Console is created on an arbitrary UCS server system as part of the UCS domain replication.

Provision for accessing clients can occur via CIFS (supported by Windows/Linux clients) and/or NFS (primarily supported by Linux/Unix). The NFS shares managed in Univention Management Console can be mounted by clients both via NFSv3 and via NFSv4.

If a file share is deleted on a server, the shared files in the directory are preserved.

To be able to use access control lists on a share, the underlying Linux file system must support POSIX ACLs. In UCS the file systems `ext3`, `ext4` and `XFS` support POSIX ACLs. The Samba configuration also allows storing DOS file attributes in extended attributes of the Unix file system. To use extended attributes, the partition must be mounted using the mount option `user_xattr`.

## 12.1. Access rights to data in shares

Feedback

Access permissions to files are managed in UCS using users and groups. All the file servers in the UCS domain access identical user and group data via the LDAP directory.

Three access rights are differentiated per file: read, write and execute. Three access rights also apply per directory: read and write are the same; the execute permission here refers to the permission to enter a directory.

Each file/directory is owned by a user and a group. The three permission outlined above can be applied to the user owner, the owner group and all others.

If the *setuid* option is set for an executable file, it can be run by users with the privileges of the owner of the file.

If the *setgid* option is set for a directory, files saved there inherit the directory's owner group. If further directories are created, they also inherit the option.

If the *sticky bit* option is enabled for a directory, files in this directory can only be deleted by the owner of the file or the root user.

Access control lists allow even more complex permission models. The configuration of ACLs is described in SDB 1042.

In the Unix permission model - and thus under UCS - write permission is not sufficient to change the permissions of a file. This is limited to the owner/owner group of a file. In contrast, under Microsoft Windows all users with write permissions also have the permission to change the permissions. This scheme can be adjusted for CIFS shares (see Section 12.2).

Only initial users and access permissions are assigned when a directory share is created. If the directory already exists, the permissions of the existing directory are adjusted.

http://sdb.univention.de/1042

Changes to the permissions of a shared directory performed directly in the file system are not forwarded to the LDAP directory. If the permissions/owners are edited within Univention Management Console, the changes in the file system are overwritten. Settings to the root directory of a file share should thus only be set and edited with Univention Management Console. Additional adjustment of the access permissions of the subordinate directories are then performed via the accessing clients, e.g., via Windows Explorer, or directly via command line commands on the file server.

The *homes* share plays a special role within Samba. This share is used for sharing the home directories of the users. This share is automatically converted to the user's home directory. Samba therefore ignores the rights assigned to the share, and uses the rights of the respective home directory instead.

## 12.2. Management of shares in UMC

Feedback

File shares are managed in the UMC module *Shares* (see Section 4.4).

When adding/editing/deleting a share, it is entered, modified or removed in the `/etc/exports` file and/or the Samba configuration.

**Figure 12.1. Creating a share in UMC**

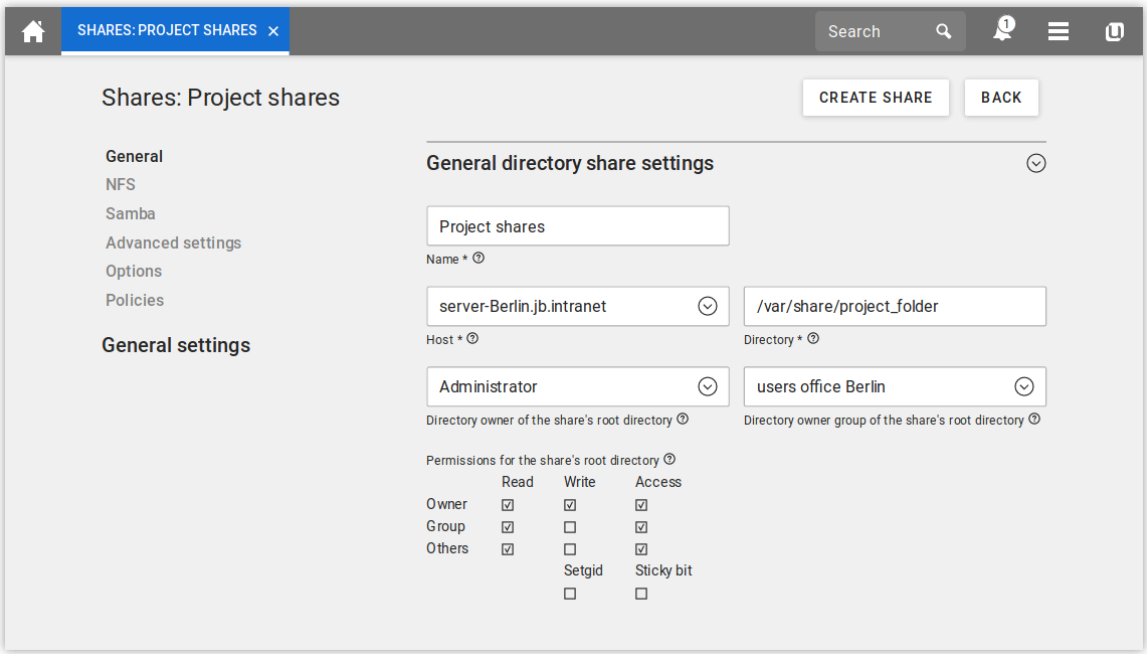



**Table 12.1. 'General' tab**

| Attribute | Description |
|---|---|
| Name | The name of the share is to be entered here. The name must be composed of letters, numerals, full stops or blank spaces and must begin and end with a letter or numeral. |
| Host | The server where the share is located. All of the domain controller master/backup/slave computers and member servers entered in the LDAP directory for the domain are available for selection which are entered in a DNS forward lookup zone in the LDAP directory. |
| Directory | The absolute path of the directory to be shared, without quotation marks (this also applies if the name includes special characters such as spaces). |

| Attribute | Description |
|---|---|
| | If the directory does not exist, it will be created automatically on the selected server.<br><br>If the Univention Configuration Registry variable `listener/shares/rename` is set to `yes`, the contents of the existing directory are moved when the path is modified.<br><br>No shares can be created in and below `/proc`, `/tmp`, `/root`, `/dev` and `/sys` and no files can be moved there. |
| Directory owner of the share's root directory | The user to whom the root directory of the share should belong, see Section 12.1. |
| Directory owner group of the share's root directory | The group to whom the root directory of the share should belong, see Section 12.1. |
| Permissions for the share's root directory | The read, write and access permissions for the root directory of the share, see Section 12.1. |

**Table 12.2. 'NFS' tab**

| Attribute | Description |
|---|---|
| NFS write access | Allows NFS write access to this share; otherwise the share can only be used in read-only mode. |
| Subtree checking | If only one subdirectory of a file system is exported, the NFS server has to check whether an accessed file is located on the exported file system and in the exported path, each time access is made. Path information is passed on to the client for this check. Activating this function might cause problems if a file opened by the client, is renamed. |
| Modify user ID for root user (root squashing) | In the NFS standard procedure, identification of users is achieved via user IDs. To prevent a local root user from working with root permissions on other shares, root access can be redirected. If this option is activated, access operations are executed as user `nobody`.<br><br>The local group `staff`, which is by default empty, owns privileges which come quite close to `root` permissions, yet this group is not considered by the redirection mechanism. This fact should be borne in mind when adding users to this group. |
| NFS synchronization | The synchronization mode for the share. The `sync` setting is used to write data directly on the underlying storage device. The opposite setting - `async` - can improve performance but also involves the risk of data loss if the server is shut down incorrectly. |
| Only allow access for these hosts, IP addresses or networks | By default, all hosts are permitted access to a share. In this select list, host names and IP addresses can be included, to which the access to the share is to be restricted. For example, access to a share containing mail data could be restricted to the mail server of the domain. |

**Table 12.3. 'Samba' tab**

| Attribute | Description |
|---|---|
| Samba name | The NetBIOS name of the share. This is the name under which the share is displayed on Windows computers in the network environment. When |

| Attribute | Description |
|---|---|
| | adding a directory share, Univention Management Console adopts the name entered in the **Name** field of the **General** tab as the default. |
| Samba write access | Permits write access to this share. |
| Show in Windows network environment | Specifies whether the share in question is to show up on Windows clients within the network environment. |
| Allow anonymous read-only access with a guest user | Permits access to this share without a password. Every access is carried out by means of the common guest user `nobody`. |
| MSDFS root | This option is documented in Section 12.3. |
| Users with write access may modify permissions | If this option is activated, all users with write permission to a file are allowed to change permissions, ACL entries, and file ownership rights, see Section 12.1. |
| Hide unreadable files/directories | If this option is activated, all files which are nonreadable for the user due to their file permissions, will be hidden. |
| VFS Objects | Virtual File System (VFS) modules are used in Samba for performing actions before an access to the file system of a share is made, e.g., a virus scanner which stores every infected file accessed in the share in quarantine or server-side implementation of recycle bin deletion of files. |

**Table 12.4. 'Samba permissions' tab (advanced settings)**

| Attribute | Description |
|---|---|
| Force user | This username and its permissions and primary group is used for performing all the file operations of accessing users. The username is only used once the user has established a connection to the Samba share by using his real username and password. A common username is useful for using data in a shared way, yet improper application might cause security problems. |
| Force group | A group which is to be used by all users connecting with this share, as their primary group. Thereby, the permissions of this group automatically apply as the group permissions of all these users. A group registered here has a higher priority than a group which was assigned as the primary group of a user via the **Force user** entry field.<br><br>If a + sign is prefixed to the group name, then the group is assigned as a primary group solely to those users which are already members of this group. All other users retain their primary groups. |
| Valid users or groups | Names of users or groups which are authorized to access this Samba share. To all other users, access is denied. If the field is empty, all users may access the share - if necessary after entering a password. This option is useful for securing access to a share at file server level beyond the file permissions.<br><br>The entries are to be separated by spaces. The special characters @, + and & can be used in connection with the group name for assigning certain permissions to the users of the stated group for accessing the Samba share:<br><br>◦ A name beginning with the character @ will first be interpreted as a NIS Netgroup. Should no NIS Netgroup of this name be found, the name will be considered as a UNIX group. |

| Attribute | Description |
|---|---|
| | ◦ A name beginning with the character + will be exclusively considered as a UNIX group, a name beginning with the character & will be exclusively considered as a NIS Netgroup.<br><br>◦ A name beginning with the characters +&, will first be interpreted as a UNIX group. Should no UNIX group of this name be found, the name will be considered as a NIS Netgroup. The characters &+ as the beginning of a name correspond to the character @. |
| Invalid users or groups | The users or groups listed here cannot access the Samba share. The syntax is identical to the one for valid users. If a user or group is included in the list of valid users and unauthorized users, access is denied. |
| Restrict write access to these users/ groups | Only the users and groups listed here have write permission for the corresponding share. |
| Allowed hosts/networks | Names of computers which are authorized to access this Samba share. All other computers are denied access. In addition to computer names, it is also possible to specify IP or network addresses, e.g., **`192.0.2.0/255.255.255.0`**. |
| Denied hosts/networks | The opposite to the authorized computers. If a computer appears in both lists, the computer is permitted to access the Samba share. |
| NT ACL support | If this option is activated, Samba will try to show POSIX ACLs under Windows, and to adopt changes to the ACLs, which were performed under Windows, for the POSIX ACLs.<br><br>If this option is not set, existing POSIX ACLs are effective but not shown under Windows, and consequently cannot be changed under Windows. |
| Inherit ACLs | When activating this option, each file created in this share will inherit the ACL (Access Control List) of the directory where the file was created. |
| Create files/directories with the owner of the parent directory | When activating this option, each newly created file will not be assigned of the user who created the file, but to the owner of the superior directory instead. |
| Create files/directories with permissions of the parent directory | When activating this option, for each file or directory created in this share, the UNIX permissions of the superior directory will automatically be adopted. |

If a new file is created on a Samba server from a Windows client, the file permissions will be set in several steps:

1. First, only the DOS permissions are translated into UNIX permissions.

2. Then the permissions are filtered via the **Filemode**. UNIX permissions which are marked in **File mode**, are the only ones preserved. Permissions not set here, will be removed. Thus, the permissions have to be set as UNIX permissions and in **File mode** in order to be preserved.

3. In the next step, the permissions under **Force file mode** are added. As a result, the file will have all the permissions set after step 2 or under **Force file mode**. This means, permissions marked under **Force file mode** are set in any case.

Accordingly, a newly created directory will initially be assigned the same permissions as that which are set as UNIX permissions and in **Directory mode** at the same time. Then these permissions are completed by those marked under **Force directory mode**.

In a similar way, the security settings are adopted for existing files and directories the permissions of which are edited under Windows:

Only those permissions can be changed under Windows, which are marked in **Security mode** or in **Directory security mode**. Once this is done, the permissions marked under **Force security mode** of under **Force directory security mode** are set in any case.

Thus, the parameters **File mode** and **Force file mode**, or **Directory mode** and **Force directory mode** are applied during the creation of a file or directory, while the parameters **Security mode** and **Force Security Mode** or **Security directory mode** and **Force security directory mode** are applied when changing permissions.

## Note

The security settings only relate to the access via Samba.

The user on the Windows side does not receive any notification of the fact that the file or directory authorizations might by changed according to the Samba settings on this tab.

### Table 12.5. 'Samba extended permissions' tab (advanced settings)

| Attribute | Description |
|---|---|
| File mode | The permissions Samba is to adopt when creating a file, provided they are set under Windows. |
| Directory mode | The permissions Samba is to adopt when creating a directory, provided they are set under Windows. |
| Force file mode | The permissions Samba is to set in any case when creating a file, irrespective of whether they are set under Windows or not. |
| Force directory mode | The permissions Samba is to set in any case when creating a directory, irrespective of whether they are set under Windows or not. |
| Security mode | The file permissions to which Samba is to permit changes made from Windows side. |
| Directory security mode | The directory authorizations to which Samba is to permit changes made from Windows side. |
| Force security mode | The permissions Samba is to set in any case (irrespective of whether they are set under Windows or not), if file permissions are changed from Windows side. |
| Force directory security mode | The permissions Samba is to set in any case if directory permissions are changed from Windows side (irrespective of whether they are set under Windows or not). |

### Table 12.6. 'Samba options' tab (advanced settings)

| Attribute | Description |
|---|---|
| Locking | Locking means preventing concurrent access to a file, making an exclusive access possible. When activating this checkbox, Samba will lock the access to files on the client's request.<br><br>Deactivating this option can be useful for improving performance, yet it should generally not be set in shares with write access, since without locking, files might be corrupted due to concurrent access. |
| Blocking locks | Clients can send a lock request with a time limit for a certain area of an open file. |

| Attribute | Description |
|---|---|
| | In case Samba is unable to comply with a the lock request, and this option is activated, then Samba will - in periodical intervals until the expiry of the time limit - try to lock the requested file area. If the option is deactivated, no further attempt will be made. |
| Strict locking | If this option is activated, Samba will with each read or write access check if the file is locked, and will deny access if required. On some systems, this procedure can take a long time.<br><br>If this option is deactivated, Samba will check if the file is locked on the client's request exclusively. Well configured clients ask for a check in all important cases, so that this option is usually unnecessary. |
| Oplocks | If this option is activated, Samba will use so-called *opportunistic locks*. This can improve the speed of file access considerably. However, the option permits clients local caching of files on a large scale. In unreliable networks it might therefore be necessary to do without Oplocks. |
| Level 2 Oplocks | When activating this option, Samba will support an extended form of Oplocks, the so-called *opportunistic read-only locks* or *Level 2 Oplocks*. Windows clients receiving a read/write Oplock for a file can then scale down this Oplock to a read-only Oplock instead of having to abandon the Oplock completely as soon as a second client opens the file. All clients supporting Level 2 Oplocks, will then cache read access processes to the file exclusively. Should one of the clients write to the file, all the other clients will be asked to abandon their Oplocks, and to empty their caches.<br><br>It is recommended to activate this option to speed up access to files which are normally not written to (e.g. programs / executable files).<br><br>**Note**<br><br>If kernel Oplocks are supported, Level 2 Oplocks will not be allowed, even if the option is activated. Only if the checkbox **Oplocks** is also ticked, this option will become valid. |
| Fake Oplocks | When activating this option, Samba will allow all Oplock requests irrespective of the number of clients having access to a file. This method considerably improves performance, and is useful for shares which can only be accessed for reading (e.g. CD-ROMs), or where it is ensured that there can never occur a situation when several clients make access at the same time.<br><br>If it cannot be excluded that several clients make reading and writing access to a file at the same time, this option should not be activated, since it may cause data loss. |
| Block size | The block size in bytes in which unoccupied disk space is to be reported to the clients. By default, this size is defined as 1024 bytes. |
| Client-side caching policy | This option specifies in which way the clients are to cache the files of this share offline. The available alternatives are *manual*, *documents*, *programs*, and *disable*. |
| Hide files | Files and directories to be accessed under Windows, yet not to be visible. Such files or directories are assigned the DOS attribute *hidden*. |

| Attribute | Description |
|---|---|
| | When entering the names of files and directories, upper and lower case letters are to be differentiated. Each entry is to be separated from the next by a slash. Since the slash can thus not be used for structuring path names, the input of path names is not possible. All files and directories of this name within the share will be hidden. The names may include spaces and the wildcards * and ?.<br><br>As an example, */.*/test/* hides all files and directories the names of which begin with a dot, or which are called test.<br><br>**Note**<br><br>Entries in this field have an impact on the speed of Samba since every time particular contents of the share are to be displayed, all files and directories have to be checked according to the active filters. |
| Postexec script | A script or command which is to be executed on the server if the connection to this share is finished. |
| Preexec script | A script or command which is to be executed on the server each time a connection to this share is established. |

**Table 12.7. 'Samba custom settings' tab (advanced settings)**

| Attribute | Description |
|---|---|
| Custom share settings | Apart from the properties which can, as a standard feature, be configured in a Samba share, this setting makes it possible to define further arbitrary Samba settings within the share. A list of available options can be obtained by the command `man smb.conf`. In **Key** the name of the option is to be entered, and in the **Value** field the value to be set. Double entries of configuration options are not checked.<br><br>**Caution**<br><br>The definition of extended Samba settings is only necessary in very special cases. The options should be thoroughly checked since they might have security-relevant effects. |

**Table 12.8. 'NFS custom settings' tab (advanced settings)**

| Attribute | Description |
|---|---|
| Custom NFS share settings | Apart from the properties in the **NFS** tab, this setting makes it possible to define further arbitrary NFS settings for the share. A list of available options can be obtained by the command `man 5 exports`. Double entries of configuration options are not checked.<br><br>**Caution**<br><br>The definition of extended NFS settings is only necessary in special cases. The options should be thoroughly checked since they might have security-relevant effects. |

**Table 12.9. '(Options)' tab**

| Attribute | Description |
|---|---|
| Export for Samba clients | This option defines whether the share is to be exported for Samba clients. |
| Export for NFS clients | This option defines whether the share is to be exported for NFS clients. |

## 12.3. Support for MSDFS

Feedback

The Microsoft Distributed File System (MSDFS) is a distributed file system which makes it possible to access shares spanning several servers and paths as a virtual directory hierarchy. The load can then be distributed across several servers.

Setting the **MSDFS Root** option for a share (see Section 12.2) indicates that the shared directory is a share which can be used for the MSDFS. References to other shares are only displayed in such an MSDFS root, elsewhere they are hidden.

To be able to utilize the functions of a distributed file system, the Univention Configuration Registry variable `samba/enable-msdfs` has to be set to `yes` on a file server. Afterwards Samba has to be restarted.

For creating a reference named `tofb` from server `sa` within the share `fa` to share `fb` on the server `sb`, the following command has to be executed in directory `fa`:

```
ln -s msdfs:sb\\fb tofb
```

This reference will be displayed on every client capable of MSDFS (e.g. `Windows 2000` and `Windows XP`) as a regular directory.

### Caution

Only restricted user groups should have write access to root directories. Otherwise, it would be possible for users to redirect references to other shares, and intercept or manipulate files. In addition, paths to the shares, as well as the references are to be spelled entirely in lower case. If changes are made in the references, the concerned clients have to be restarted. Further information on this issue can be found in the Samba documentation [samba3-howto-chapter-20] in the chapter 'Hosting a Microsoft Distributed File System Tree'.

## 12.4. Configuration of file system quota

Feedback

UCS allows the limiting of the storage space available to a user on a partition. These thresholds can be set as either a quantity of storage space (e.g., 500 MB per user) or a maximum number of files without a defined size limit.

Two types of thresholds are differentiated:

- The *hard limit* is the maximum storage space a user can employ. If it is attained, no further files can be saved.
- If the *soft limit* is attained - which must be smaller than the hard limit - and the storage space used is still below the hard limit, the user is given a grace period of seven days to delete unused data. Once seven days have elapsed, it is no longer possible to save or change additional files. A warning is displayed to users who access a file system with an exceeded quota via CIFS (the threshold is based on the soft limit).

If a quota value of *0* has been configured, it is evaluated as an unlimited quota.

Quotas can either be defined via the UMC module **Filesystem quotas** or a policy for shares, see Section 12.4.2.

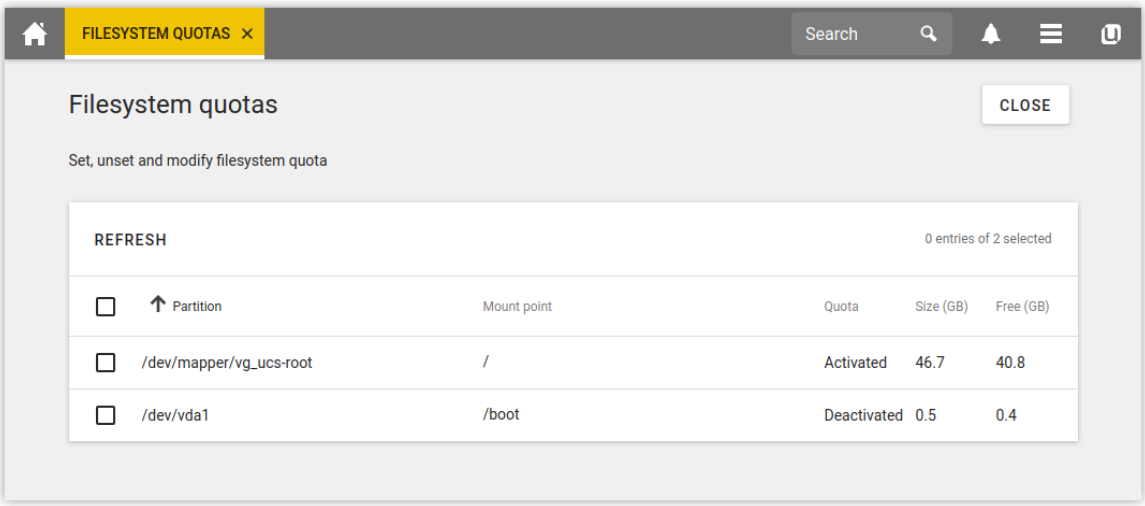
File system quotas can only be applied on partitions with the file systems `ext2`, `ext3`, `ext4` and `XFS`. Before filesystem quotas can be configured, the use of file system quotas needs to be activated per partition, see Section 12.4.1.

## 12.4.1. Activating filesystem quota

Feedback

In the UMC module **Filesystem quotas**, all the partitions are listed on which quotas can be set up. Only partitions are shown which are currently mounted under a mount point.

**Figure 12.2. Managing quota in the UMC**




The current quota status (activated/deactivated) is shown and can be changed with **Activate** and **Deactivate**.

If quota has been activated on a `XFS` root-partition, the system has to be rebooted.

## 12.4.2. Configuring filesystem quota

Feedback

Quotas can either be defined via the UMC module **Filesystem quotas** or a policy for shares, see Section 4.6. The configuration through a policy allows setting a default value for all users, while the UMC module allows easy configuration of user-specific quota values.

The user-specific quota settings can be configured with the UMC module **Filesystem quotas**. The permitted storage quantities can be set with the pencil symbol for all enabled partitions. All the settings are set user-specifically. **Add** can be used to set the thresholds for soft and hard limits for a user.

The quota settings can also be set with a **User quota** share policy. The settings apply for all users of a share; it is not possible to establish different quota limits for different users within one policy.

Quota settings that are applied via a quota policy are by default only applied once to the filesystem. If the setting is changed, it will not be applied automatically on the next user login. To inherit changed quota values, the option *Reapply settings on every login* can be activated at the quota policy.

Quota policies can only be used on partitions for which the quota support is enabled in the UMC module, see Section 12.4.1.

### Note

Filesystem quotas always apply to a full partition. Even if the policies are defined for shares, they are used on complete partitions. If, for example, three shares are provided on one server which are all

saved on the separate `/var/` partition and three different policies are configured and used, the most restrictive setting applies for the complete partition. If different quotas are used, it is recommended to distribute the data over individual partitions.

## 12.4.3. Evaluation of quota during login

Feedback

The settings defined in the UCS management system are evaluated and enabled during login to UCS systems by the tool `univention-user-quota` run in the PAM stack.

If no quota are needed, the evaluation can be disabled by setting the Univention Configuration Registry variable `quota/userdefault` to `no`.

If the Univention Configuration Registry variable `quota/logfile` is set to any file name, the activation of the quotas is logged in the specified file.

## 12.4.4. Querying the quota status by administrators or users

Feedback

A user can view the quota limits defined for a system using the command `repquota -va`, e.g.:

```
*** Report for user quotas on device /dev/vdb1
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --      20       0       0              2     0     0
Administrator --       0       0  102400              0     0     0
user01    --  234472 2048000 4096000              2     0     0
user02    --       0 2048000 4096000              0     0     0

Statistics:
Total blocks: 8
Data blocks: 1
Entries: 4
Used average: 4.000000
```

Logged in users can use the `quota -v` command to view the applicable quota limits and the current utilization.

Further information on the commands can be found in the man pages of the commands.

# Chapter 13. Print services

13.1. Introduction ..... 239
13.2. Installing a print server ..... 239
13.3. Setting the local configuration properties of a print server ..... 240
13.4. Creating a printer share ..... 240
13.5. Creating a printer group ..... 243
13.6. Administration of print jobs and print queues ..... 245
13.7. Generating PDF documents from print jobs ..... 246
13.8. Mounting of print shares in Windows clients ..... 247
13.9. Integrating additional PPD files ..... 251

## 13.1. Introduction

Feedback

Univention Corporate Server includes a print system, which can also be used to realize complex environments. Printers and printer groups can be created and configured conveniently in Univention Management Console. Extensions for cost calculation and page limitation can be installed subsequently using the print quota system.

The print services are based on *CUPS (Common Unix Printing System)*. *CUPS* manages print jobs in print queues and converts print jobs into the native formats of the connected printers. The print queues are also administrated in Univention Management Console, see Section 13.6.

All printers set up in CUPS can be directly used by UCS systems and are automatically also provided for Windows computers when Samba is used.

The technical capacities of a printer are specified in so-called PPD files. These files include for example whether a printer can print in color, whether duplex printing is possible, whether there are several paper trays, which resolutions are supported and which printer control languages are supported (e.g., PCL or PostScript).

Print jobs are transformed by CUPS with the help of filters into a format that the respective printer can interpret, for example into PostScript for a PostScript-compatible printer.

UCS already includes a wide variety of filters and PPD files. Consequently, most printers can be employed without the need to install additional drivers. The setting up of additional PPD files is described in Section 13.9.

A printer can either be connected directly to the print server locally (e.g., via the USB port or a parallel port) or communicate with a printer via remote protocols (e.g., TCP/IP compatible printers, which are connected via `IPP` or `LPD`).

Network printers with their own IP address should be registered in the computer administration of Univention Management Console as an IP managed client (see Section 3.3).

CUPS offers the possibility of defining printer groups. The included printers are used employed alternating, which allows automatic load distribution between neighboring printers.

The print quota system, which can be installed using the ***univention-printquota*** package, can be used to install an expansion for recording incurred printer costs and for limiting the number of pages to be printed. The setting and configuration is documented in the extended documentation [ext-print-doc].

Print shares from Windows systems can also be integrated in the CUPS print server, see Section 13.4.

## 13.2. Installing a print server

Feedback

A printserver can be installed from the Univention App Center with the application *Print server (CUPS)*. Alternatively, the software package ***univention-printserver*** can be installed (`univention-run-join-scripts` must be executed after installation). Additional information can be found in Section 5.6.

## 13.3. Setting the local configuration properties of a print server

Feedback

The configuration of the CUPS print server is performed via settings from the LDAP directory service and Univention Configuration Registry. If the Univention Configuration Registry variable `cups/include/local` is set to `true`, the `/etc/cups/cupsd.local.conf` file is included, in which arbitrary options can be defined. Changes in this file require `ucr commit /etc/cups/cupsd.conf` to get applied.

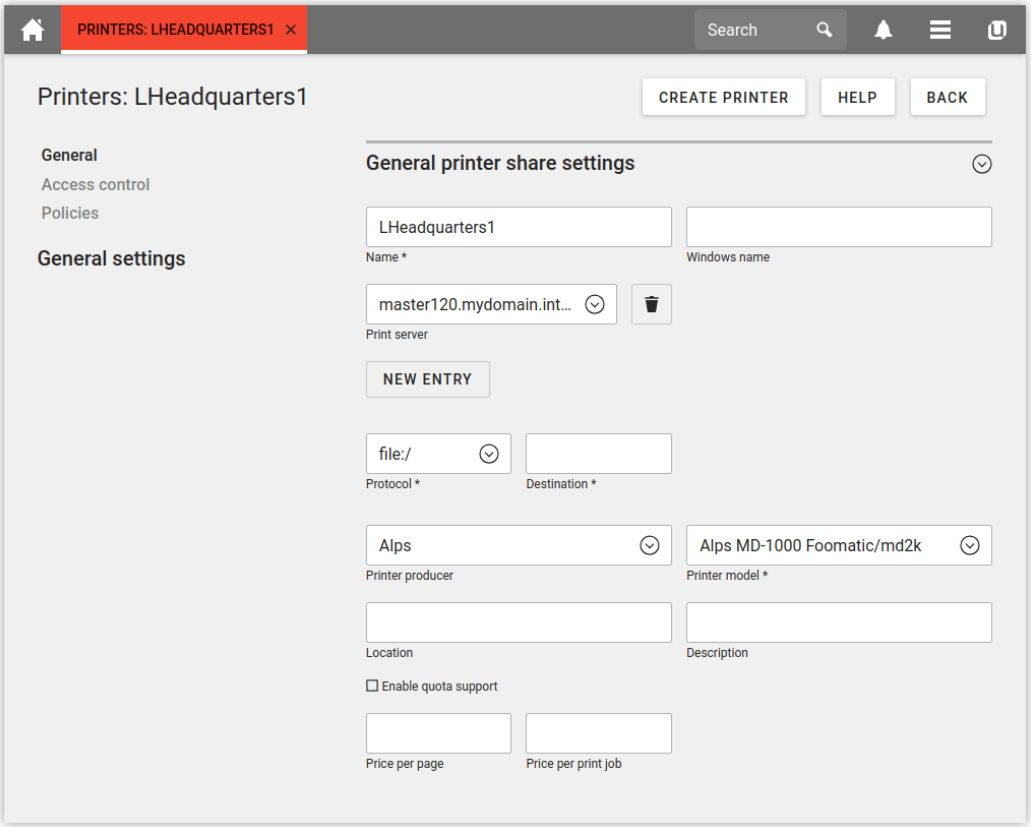
If an error occurs when working through a printer queue (e.g., because the connected printer is switched off), the further processing of the queue is stopped in the default setting. This must then be reactivated by the administrator (see Section 13.6). If the Univention Configuration Registry variable `cups/errorpolicy` is set to *retry-job*, CUPS automatically attempts to process unsuccessful print jobs again every 30 seconds.

## 13.4. Creating a printer share

Feedback

Print shares are administrated in the UMC module *Printers* with the **Printer share** object type (see Section 4.4).

**Figure 13.1. Creating a printer share**




When adding/deleting/editing a printer share, the printer is automatically configured in CUPS. CUPS does not have an LDAP interface for printer configuration, instead the `printers.conf` file is generated via a listener module. If Samba is used, the printer shares are also automatically provided for Windows clients.

**Table 13.1. 'General' tab**

| Attribute | Description |
|---|---|
| Name (*) | This input field contains the name of the printer share, which is used by CUPS. The printer appears under this name in Linux and Windows. The name may contain alphanumeric characters (i.e., uppercase and lowercase letters a to z and numbers 0 to 9) as well as hyphens and underscores. Other characters (including blank spaces) are not permitted. |
| Print server (*) | A print server manages the printer queue for the printers to be shared. It converts the data to be printed into a compatible print format when this is necessary. If the printer is not ready, the print server saves the print jobs temporarily and forwards them on to the printer subsequently. If more than one print server is specified, the print job from the client will be sent to the first print server to become available.<br><br>Only domain controllers and member servers on which the ***univention-printserver*** package is installed are displayed in the list. |
| Protocol and Destination (*) | These two input fields specify how the print server accesses the printer:<br><br>The following list describes the syntax of the individual protocols for the configuration of printers connected locally to the server:<br><br>◦ `parallel://devicefile`<br><br>Example: `parallel://dev/lp0`<br><br>◦ `socket://server:port`<br><br>Example: `socket://printer_03:9100`<br><br>◦ `usb://devicefile`<br><br>Example: `usb://dev/usb/lp0`<br><br>The following list describes the syntax of the individual protocols for the configuration of network printers:<br><br>◦ `http://server[:port]/path`<br><br>Example: `http://192.0.2.10:631/printers/remote`<br><br>◦ `ipp://server/printers/queue`<br><br>Example: `ipp://printer_01/printers/xerox`<br><br>◦ `lpd://server/queue`<br><br>Example: `lpd://192.0.2.30/bwdraft`<br><br>The `cups-pdf` protocol is used for integrating a pseudo printer, which creates a PDF document from all the print jobs. The setup is documented in Section 13.7.<br><br>The `file:/` protocol expects a file name as a target. The print job is then not sent to the printer, but instead written in this file, which can be useful for test purposes. The file is rewritten with every print job. |

| Attribute | Description |
|---|---|
| | The `smb://` protocol can be used to mount a Windows print share. For example, to integrate the `laser01` printer share from Windows system `win01`, `win01/laser01` must be specified as destination. The manufacturer and model must be selected according to the printer in question. The print server uses the printer model settings to convert the print jobs where necessary and send these directly to the URI `smb://win01/laser01`. No Windows drivers are used in this.<br><br>Independent of these settings, the printer share can be mounted by other Windows systems with the corresponding printer drivers. |
| Manufacturer | When the printer manufacturer is selected, the *Printer model* selection list updates automatically. |
| Printer model (*) | This selection list shows all the printers PPD files available for the selected manufacturer. If the required printer model is not there, a similar model can be selected and a test print used to establish correct function. Section 13.9 explains how to expand the list of printer models. |
| Samba name | A printer can also be assigned an additional name by which it can be reached from Windows. Unlike the CUPS name (see **Name**), the Samba name may contain blank spaces and umlauts. The printer is then available to Windows under both the CUPS name and the Samba name.<br><br>Using a Samba name in addition to the CUPS name is practical, for example, if the printer was already in use in Windows under a name which contains blank spaces or umlauts. The printer can then still be reached under this name without the need to reconfigure the Windows computers. |
| Enable quota support | If quota were activated for this printer, the quota settings on the **[Print Quota]** policy apply.<br><br>The print quota system needs to be installed for this, see [ext-print-doc]. |
| Price per page | The user is charged the value given in this input field for every page printed. The incurred costs are summarized in the user's account and used for the accurate calculation of print costs. If no value is specified, print costs will not be calculated.<br><br>The print quota system needs to be installed for this. |
| Price per print job | The user is charged the value given in this input field for every print job. The incurred costs are summarized in the user's account and used for the accurate calculation of print costs. If no value is specified, print costs will not be calculated.<br><br>The print quota system needs to be installed for this. |
| Location | This data is displayed by some applications when selecting the printer. It can be filled with any text. |
| Description | This is displayed by some applications when selecting the printer. It can be filled with any text. |

**Table 13.2. 'Access Control' tab**

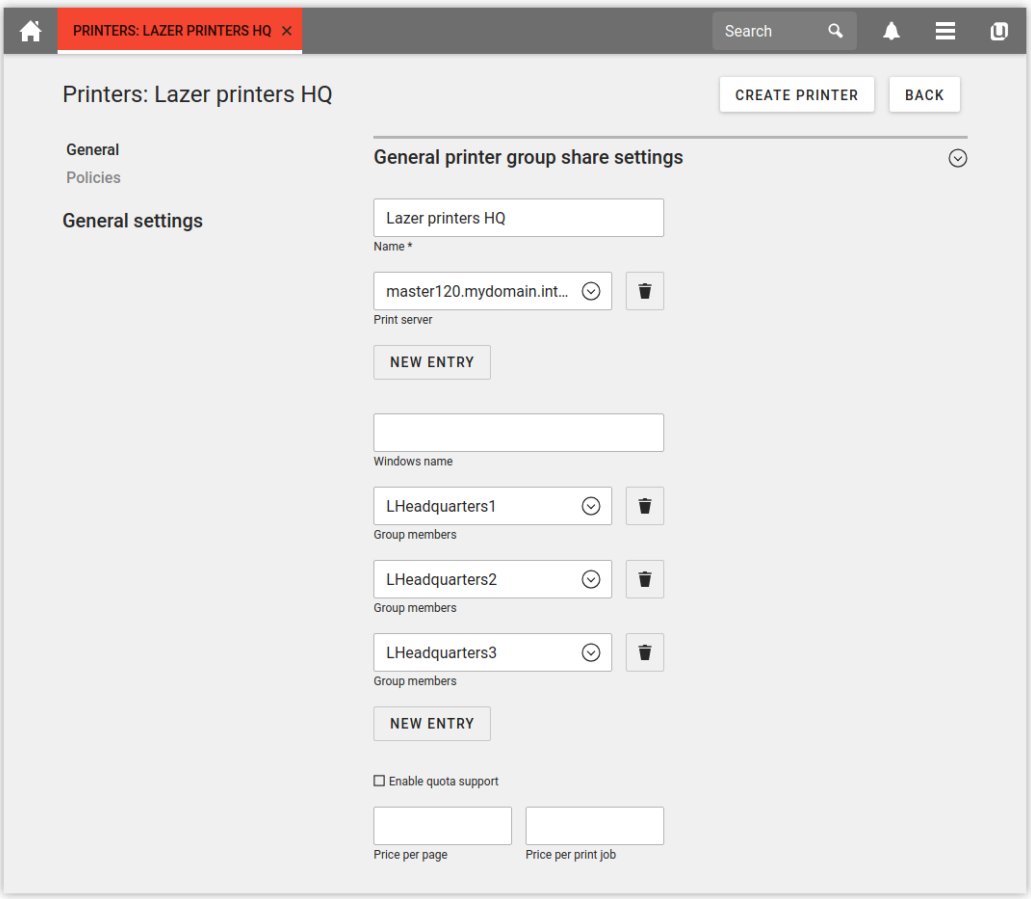
| Attribute | Description |
|---|---|
| Access control | Access rights for the printer can be specified here. Access can be limited to certain groups or users or generally allowed and certain groups or users blocked specifically. As standard, access is available for all groups and users. These rights are also adopted for the corresponding Samba printer shares, so that the same access rights apply when printing via Samba as when printing directly via CUPS.<br><br>This access control is useful for the management of printers spread across several locations, so that the users at location A do not see the printers of location B. |
| Allowed/denied users | This lists individual users for whom access should be controlled. |
| Allowed/denied groups | This lists individual groups for whom access should be controlled. |

## 13.5. Creating a printer group

Feedback

CUPS offers the possibility to group printers into classes. These are implemented in UCS as *printer groups*. Printer groups appear to clients as normal printers. The aim of such a printer group is to create a higher availability of printer services. If the printer group is used to print, the job is sent to the first printer in the printer group to become available. The printers are selected based on the round robin principle so that the degree of utilization is kept uniform.

**Figure 13.2. Configuring a printer group**




A printer group must have at least one printer as a member. Only printers from the same server can be members of the group.

## Caution

The possibility of grouping printers shares from different printer servers in a printer group makes it possible to select printer groups as members of a printer group. This could result in a printer group adopting itself as a group member. This must not be allowed to happen.

Printer groups are administrated in the UMC module *Printers* with the **Printer share** object type (see Section 4.4).

**Table 13.3. 'General' tab**

| Attribute | Description |
|---|---|
| Name (*) | This input field contains the names of the printer group share, which is used by CUPS. The printer group appears under this name in Linux and Windows.<br><br>The name may contain alphanumeric characters (i.e., uppercase and lowercase letters a to z and numbers 0 to 9) as well as hyphens and underscores. Other characters (including blank spaces) are not permitted. |

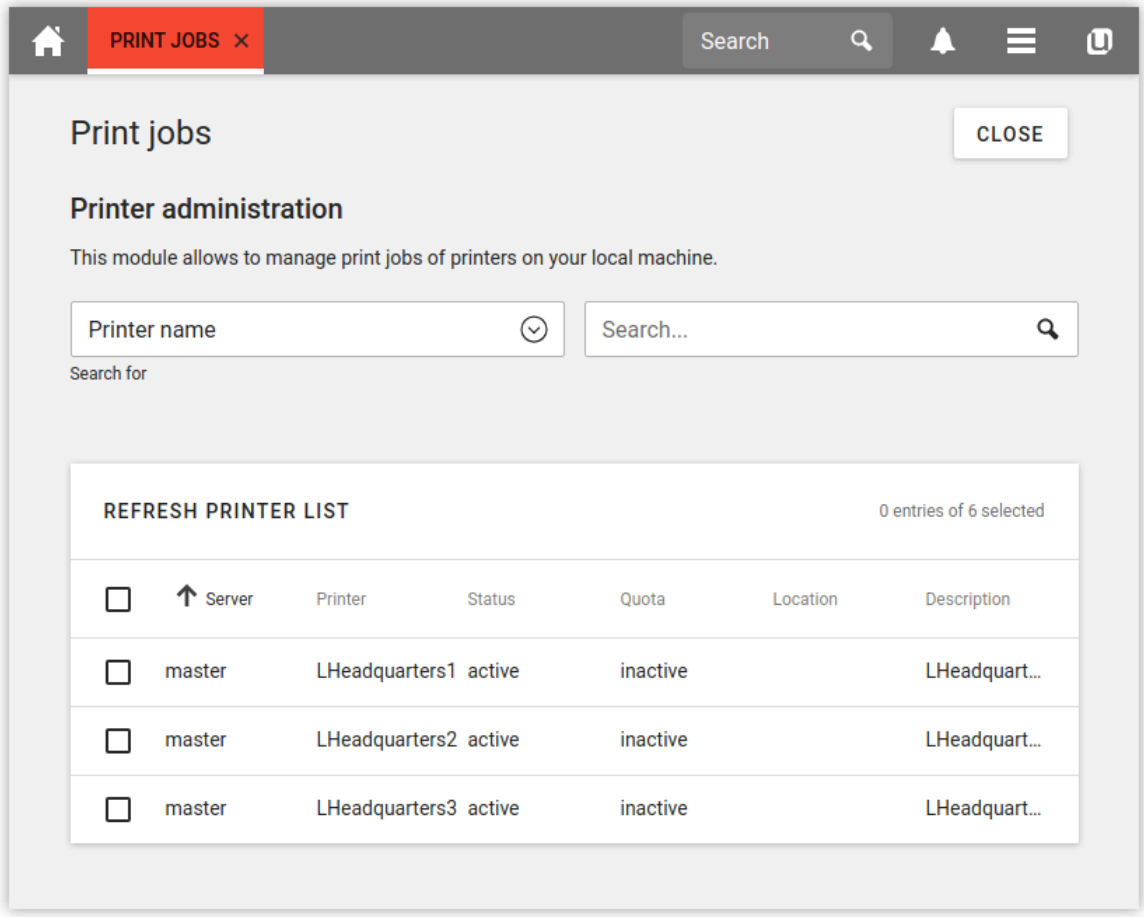
| Attribute | Description |
|---|---|
| Print server (*) | A range of print servers (spoolers) can be specified here to expand the list of printers available for selection. Printers which are assigned to the servers specified here can then be adopted in the **Group members** list from the selection arranged below them. |
| Samba name | A printer group can also be assigned an additional name by which it can reached from Windows. Unlike the CUPS name (see *Name*), the Samba name may contain blank spaces and umlauts. The printer is then available to Windows under both the CUPS name and the Samba name.<br><br>Using a Samba name in addition to the CUPS name is practical, for example, if the printer group was already in use in Windows under a name which contains blank spaces or umlauts. The printer group can then still be reached under this name without the need to reconfigure the Windows computers. |
| Group members | This list is used to assign printers to the printer group. |
| Enable quota support | If quota were activated for this printer group, the quota settings on the **[Print Quota]** tab apply.<br><br>The print quota system needs to be installed for this, see [ext-print-doc]. |
| Price per page | The user is charged the value given in this input field for every page printed. The incurred costs are summarized in the user's account and used for the accurate calculation of print costs. If no value is specified, print costs will not be calculated. |
| Price per print job | The user is charged the value given in this input field for every print job. The incurred costs are summarized in the user's account and used for the accurate calculation of print costs. If no value is specified, print costs will not be calculated. |

## 13.6. Administration of print jobs and print queues

Feedback

The UMC module **Printer Administration** allows you to check the status of the connected printers, restart paused printers and remove print jobs from the queues on printer servers.

**Figure 13.3. Printer administration**




The start page of the module contains a search mask with which the available printers can be selected. The results list displays the server, name, status, print quota properties, location and description of the respective printer. The status of more than one printer can be changed simultaneously by selecting the printers and running either the **deactivate** or **activate** function.

The configuration of the print quota settings is documented in the extended documentation [ext-print-doc].

Clicking on the printer name displays details of the selected printer. The information displayed includes a list of the print jobs currently in the printer queue. These print jobs can be deleted from the queue by selecting the jobs and running the **[Delete]** function.

## 13.7. Generating PDF documents from print jobs

Feedback

Installing the ***univention-printserver-pdf*** package expands the print server with a special **cups-pdf** printer type, which converts incoming print jobs into PDF documents and add them in a specified directory on the printer server where they are readable for the respective user. After the installation of the package, `univention-run-join-scripts` must be run.

The **cups-pdf:/** protocol must be selected when creating a PDF printer in Univention Management Console (see Section 13.4); the destination field remains empty.

*PDF* must be selected as **Printer producer** and **Generic CUPS-PDF Printer** as **Printer model**.

The target directory for the generated PDF documents is set using the Univention Configuration Registry variable `cups/cups-pdf/directory`. As standard it is set to `/var/spool/cups-pdf/%U` so that cups-pdf uses a different directory for each user.

Print jobs coming in anonymously are printed in the directory specified by the Univention Configuration Registry variable `cups/cups-pdf/anonymous` (standard setting: `/var/spool/cups-pdf/`).

In the default setting, generated PDF documents are kept without any restrictions. If the Univention Configuration Registry variable `cups/cups-pdf/cleanup/enabled` is set to `true`, old PDF print jobs are deleted via a Cron job. The storage time in days can be configured using the Univention Configuration Registry variable `cups/cups-pdf/cleanup/keep`.

## 13.8. Mounting of print shares in Windows clients

Feedback

The printer shares set up in Univention Management Console can be added as network printers on Windows systems. This is done via the Control Panel under **Add a device -> Add a printer**. The printer drivers need to be set up during the first access. If the drivers are stored on the server side (see below), the drivers are assigned automatically.

Printer shares are usually operated using the Windows printer drivers provided. The network printer can alternatively be set up on the Windows side with a standard PostScript printer driver. If a color printer should be accessed, a driver for a PostScript-compatible color printer should be used on the Windows side, e.g., *HP Color LaserJet 8550*.

### Caution

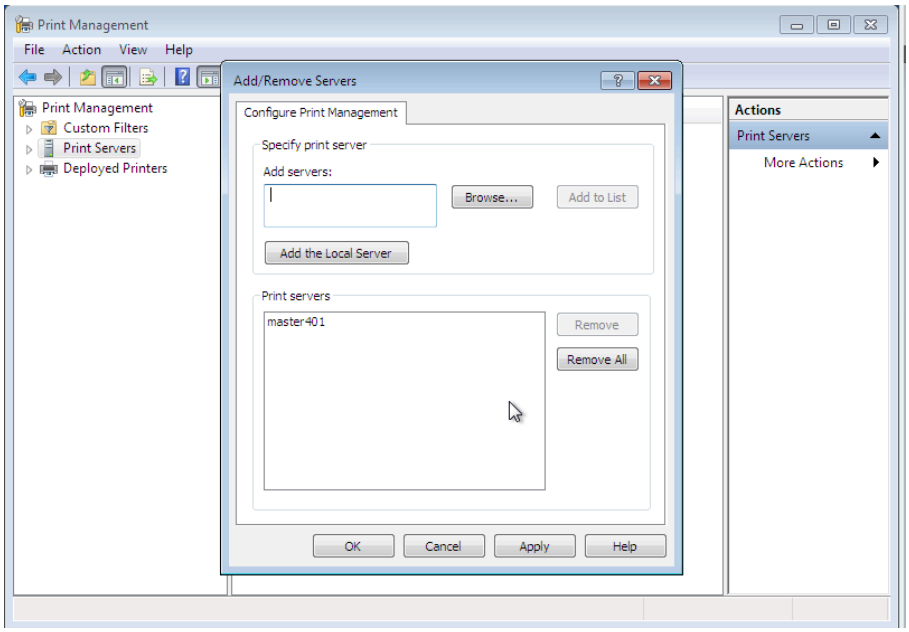
The printer can only be accessed by a regular user when he has local permissions for driver installation or the respective printer drivers were stored on the printer server. If this is not the case, Windows may issue an error warning that the permissions are insufficient to establish a connection to the printer.

Windows supports a mechanism for providing the printer drivers on the print server (*Point 'n' Print*). The following guide describes the provision of the printer drivers in Windows 7 for a print share configured in the UMC. Firstly, the printer drivers must be stored on the print server. There are a number of pitfalls in the Windows user wizard, so it is important to follow the individual steps precisely.

1. Firstly, the printer drivers must be downloaded from the manufacturer's website. If you are using an environment in which 64-bit installations of Windows are used, you will need both versions of the drivers (32 and 64 bit). The `INF` files are required.

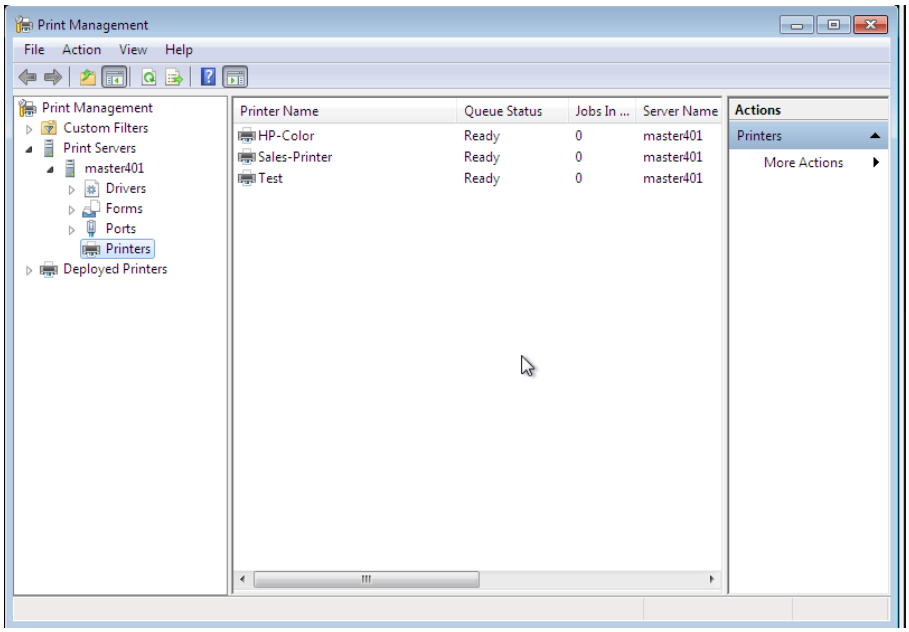
2. Now you need to start the **printmanagement.msc** program. Clicking on **Add/remove server** in the **Action** menu item allows you to add another server. The name of the printer server needs to be entered in the **Add server** input field.

**Figure 13.4. Add printer server**




3. The newly added printer server should now be listed in the print management program. Clicking on **Printers** displays the printer shares currently set up on the printer server.

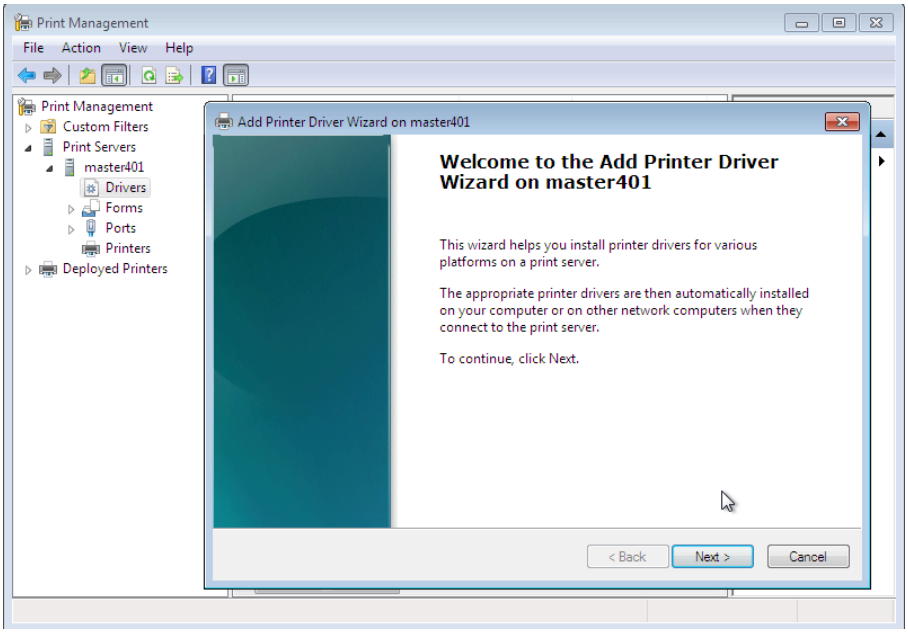
**Figure 13.5. Printer list**




4. Clicking on **Drivers** lists the saved printer drivers. Clicking on **Add driver** in the **Action** menu item opens the dialogue window for the driver installation. We recommend downloading the printer drivers directly from the manufacturer and selecting them during the driver installation. If you are using an environment containing 64-bit versions of Windows, start by performing a check to see if the Univention Configuration Registry variable `samba/spoolss/architecture` is set to `Windows x64` on the UCS Samba system. If this is not the case, both the 32-bit and the 64-bit drivers must be uploaded; if your domain

only uses 64-bit Windows systems, the 32-bit driver can be ignored. The drivers for the different Windows architectures can be uploaded one after the other or together. If both driver architectures are selected for uploading at the same time, the 64-bit driver should be selected first in the subsequent file selection window. Once Windows has uploaded these files to the server, it asks for the location of the 32-bit drivers again. They are then also uploaded to the server.

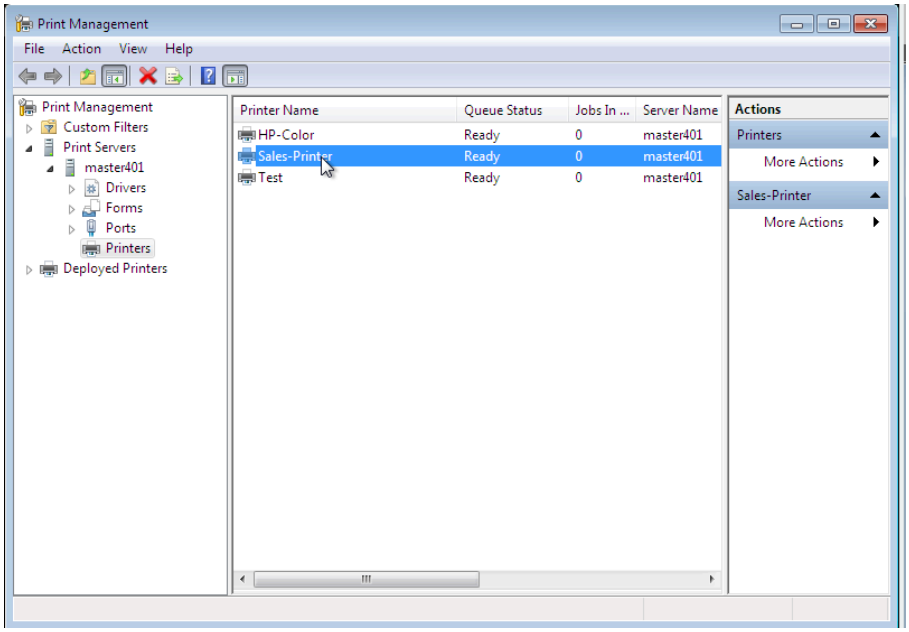
**Figure 13.6. Driver installation**




5. After these steps the drivers are stored in the directory `/var/lib/samba/drivers/` on the print server.

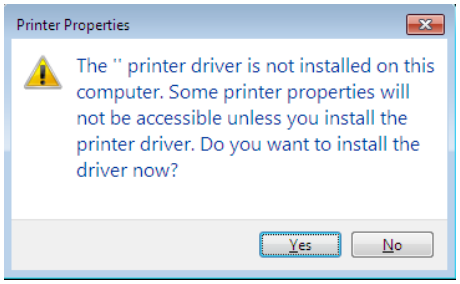
6. The print share now needs to be linked to the uploaded printer driver. To do so, the list of the printers available on the printer server is opened in the **printmanagement.msc** program. The properties can be listed there by double-clicking on the **printer**.

**Figure 13.7. Selecting a printer**




7. If no printer driver is saved, a message is displayed saying that there is no printer driver installed. The prompt to install the driver should be closed with **No** here.

**Figure 13.8. Error message on first access**




8. The uploaded driver now needs to be selected from the dropdown menu under **Drivers** in the **Advanced** tab. Then click on **Apply** (Important: Do not click on **OK**!).

9. If the printer driver in question is being assigned to a printer for the first time, a dialogue window is shown, asking whether the printer can be trusted. This should be confirmed with **Install driver**. The printer drivers saved on the server side are now downloaded to the client. If the printer driver in question has already been downloaded to the Windows system in question in this manner before, Windows displays an error message at this point `0x0000007a`. This can simply be ignored.

10. Important: Now, instead of clicking directly on **OK**, you need to return to the **General** tab again. The old name for the printer share should still be displayed on the tab. In UCS releases earlier than UCS 4.0-1, it is possible that the Windows system has changed the name of the printer share to the name of the printer driver. If that were accepted, the printer would no longer be associated with the share! If this is the case, the name of the printer on the **General** tab (the first input field next to the stylized printer symbol) needs to be reset to the name of the print share. This can be done using the **Samba name** field configured in the print management of the UMC (or if this was left blank, use the value from **Name**). If the name has had

to be reset in this fashion, Windows then asks if you are sure that you want to change the name when **OK** is clicked. Confirm the prompt.

11. To give the Windows printer driver the opportunity to save correct standard settings for the printer, you now need to switch to the **Device settings** tab. The name of the tab differs from manufacturer to manufacturer and may also be **Settings** or even just **Configuration**. Clicking on **OK** closes the window. You can then print a test page. If Windows displays an error message here `0x00000006`, the printer settings must be checked again to see whether there is a manufacturer-specific tab called **Device settings** (or something similar). If so, it should be opened and then simply confirmed with **OK**. This closes the dialogue window and saves the printer drivers settings (`PrinterDriverData`)in the Samba registry.

12. At this point, it is also practical to make the settings for the paper size and other parameters, so that they are saved in the print share. Other Windows systems which subsequently access the print share will then find the correct settings automatically. These settings can usually be opened by clicking on the **Standard values...** button in the **Advanced** tab of the printer settings. The dialogue window which opens also varies from manufacturer to manufacturer. Typically, the settings for paper size and orientation are found on a tab called **Page set-up** or **Paper/Quality**. Once the dialogue has been confirmed by clicking on **OK**, the printer driver saves these settings (as `Default DevMode`) for the printer in the Samba registry.

## 13.9. Integrating additional PPD files

Feedback

The technical capacities of a printer are specified in so-called PPD files. These files include for example whether a printer can print in color, whether duplex printing is possible, whether there are several paper trays, which resolutions are supported and which printer control languages are supported (e.g., PCL or PostScript).

In addition to the PPD files already included in the standard scope, additional ones can be added via Univention Management Console. The PPDs are generally provided by the printer manufacturer and need to be copied into the `/usr/share/ppd/` directory on the print servers.

The printer driver lists are administrated in the UMC module **LDAP directory**. There you need to switch to the `univention` container and then to the `cups` subcontainer. Printer driver lists already exist for the majority of printer manufacturers. These can be expanded or new ones can be added.

**Table 13.4. 'General' tab**

| Attribute | Description |
|---|---|
| Name (*) | The name of the printer driver list. The name under which the list appears in the **Printer model** selection list on the **General** tab for printer shares (see Section 13.4). |
| Driver | The path to the `ppd` file or to the `/usr/share/ppd/` directory. For example, if the `/usr/share/ppd/laserjet.ppd` should be used, `laserjet.ppd` must be entered here. `gzip` compressed files (file ending `.gz`) can also be entered here. |
| Description | A description of the printer driver, under which it appears in the **Printer model** selection list on the **General** tab for printer shares. |

# Chapter 14. Mail services

14.1. Introduction ..... 253
14.2. Installation ..... 254
14.3. Management of the mail server data ..... 254
14.3.1. Management of mail domains ..... 254
14.3.2. Assignment of e-mail addresses to users ..... 255
14.3.3. Management of mailing lists ..... 255
14.3.4. Management of mail groups ..... 256
14.3.5. Management of shared IMAP folders ..... 257
14.3.6. Mail quota ..... 258
14.4. Spam detection and filtering ..... 259
14.5. Identification of viruses and malware ..... 260
14.6. Identification of Spam sources with *DNS-based Blackhole Lists* (DNSBL) ..... 260
14.7. Integration of Fetchmail for retrieving mail from external mailboxes ..... 261
14.8. Configuration of the mail server ..... 261
14.8.1. Configuration of a relay host for sending the e-mails ..... 261
14.8.2. Configuration of the maximum mail size ..... 262
14.8.3. Configuration of a blind carbon copy for mail archiving solutions ..... 262
14.8.4. Configuration of soft bounces ..... 262
14.8.5. Configuration of SMTP ports ..... 262
14.8.6. Configuration of additional checks by postscreen ..... 263
14.8.7. Custom Postfix configuration ..... 263
14.8.8. Configuring the Alias-expansion-limit ..... 264
14.8.9. Handling of mailboxes during e-mail changes and the deletion of user accounts ..... 264
14.8.10. Distribution of an installation on several mail servers ..... 265
14.8.11. Mail storage on NFS ..... 265
14.8.12. Connection limits ..... 265
14.9. Configuration of mail clients for the mail server ..... 267
14.10. Webmail and administration of e-mail filters with Horde ..... 267
14.10.1. Login and overview ..... 267
14.10.2. Web-based mail access ..... 268
14.10.3. Address book ..... 269
14.10.4. E-mail filters ..... 269

## 14.1. Introduction

Feedback

Univention Corporate Server provides mail services that users can access both via standard mail clients such as Thunderbird and via the webmail interface Horde.

Postfix is used for sending and receiving mails. In the basic installation, a configuration equipped for local mail delivery is set up on every UCS system. In this configuration, Postfix only accepts mails from the local server and they can also only be delivered to local system users.

The installation of the mail server component implements a complete mail transport via SMTP (see Section 14.2). Postfix is reconfigured during the installation of the component so that a validity test in the form of a search in the LDAP directory is performed for incoming e-mails. That means that e-mails are only accepted for e-mail addresses defined in the LDAP directory or via an alias.

The IMAP service Dovecot is also installed on the system along with the mail server component. It provides e-mail accounts for the domain users and offers corresponding interfaces for access via e-mail clients. Dovecot is preconfigured for the fetching of e-mails via `IMAP` and `POP3`. Access via POP3 can be deactivated by setting the Univention Configuration Registry variable `mail/dovecot/pop3` to `no`. The same applies to IMAP

and the Univention Configuration Registry variable `mail/dovecot/imap`. The further configuration of the mail server is performed via Univention Configuration Registry, as well (see Section 14.8).

### Note

Since Univention Corporate Server version 4.0-2 Dovecot is used as the default IMAP and POP3 server. Since Univention Corporate Server version 4.3-0 the integration of Cyrus is not available anymore.

The management of the user data of the mail server (e.g., e-mail addresses or mailing list) is performed via Univention Management Console and is documented in Section 14.3 User data are stored in LDAP. The authentication is performed using a user's primary e-mail address, i.e., it must be entered as the user name in mail clients. As soon as a primary e-mail address is assigned to a user in the LDAP directory, a listener module creates an IMAP mailbox on the mail home server. By specifying the mail home server, user e-mail accounts can be distributed over several mail servers, as well (see Section 14.8.10).

Optionally, e-mails received via Postfix can be checked for Spam content and viruses before further processing by Dovecot. Spam e-mails are detected by the classification software SpamAssassin (Section 14.4); ClamAV is used for the detection of viruses and other malware (Section 14.5).

In the default setting, e-mails to external domains are delivered directly to the responsible SMTP server of that domain. Its location is performed via the resolution of the MX record in the DNS. Mail sending can also be taken over by the relay host, e.g., on the Internet provider (see Section 14.8.1).

The Horde framework is available for web-based mail access (see Section 14.10). The UCS mail system does not offer any groupware functionality such as shared calendars or invitations to appointments. However, there are groupware systems based on UCS which integrate in the UCS management system such as Kolab, Zarafa and Open-Xchange. Further information can be found in the Univention App Center (see Section 5.3).

## 14.2. Installation

Feedback

A mail server can be installed from the Univention App Center with the application *Mail server*. Alternatively, the software package ***univention-mail-server*** can be installed. Additional information can be found in Section 5.6. A mail server can be installed on all server system roles. The use of a domain controller is recommended because of frequent LDAP accesses.

The runtime data of the Dovecot server are stored in the `/var/spool/dovecot/` directory. If this directory is on a NFS share, please read Section 14.8.11.

The webmail interface Horde can be installed via the Univention App Center (see Section 5.3).

## 14.3. Management of the mail server data

Feedback

### 14.3.1. Management of mail domains

Feedback

A mail domain is an common namespace for e-mail addresses, mailing lists and IMAP group folders. Postfix differentiates between the delivery of e-mails between local and external domains. Delivery to mailboxes defined in the LDAP directory is only conducted for e-mail address from local domains. The name of a mail domain may only be composed of lowercase letters, the figures 0-9, full stops and hyphens.

Several mail domains can be managed with UCS. The managed mail domains do not need to be the DNS domains of the server - they can be selected at will. The mail domains registered on a mail server are automatically saved in the Univention Configuration Registry variable `mail/hosteddomains`.

To ensure that external senders can also send e-mails to members of the domain, MX records must be created in the configuration of the authoritative name servers, which designate the UCS server as mail server for the domain. These DNS adjustments are generally performed by an Internet provider.

Mail domains are managed in the UMC module *Mail* with the **Mail domain** object type.

## 14.3.2. Assignment of e-mail addresses to users

Feedback

A user can be assigned three different types of e-mail addresses:

- The *primary e-mail address* is used for authentication on Postfix and Dovecot. Primary e-mail addresses must always be unique. Only one primary e-mail address can be configured for every user. It also defines the user's IMAP mailbox. If a mail home server is assigned to a user (see Section 14.8.10), the IMAP inbox is automatically created by a Univention directory listener module. The domain part of the e-mail address must be registered in Univention Management Console (see Section 14.3.1).

- E-mails to *alternative e-mail addresses* are also delivered to the user's mailbox. As many addresses can be entered as you wish. The alternative e-mail addresses do not have to be unique: if two users have the same e-mail address, they both receive all the e-mails which are sent to this address. The domain part of the e-mail address must be registered in Univention Management Console (see Section 14.3.1). To receive e-mails to alternative e-mail addresses, a user must have a primary e-mail address.

- If *forward e-mail addresses* are configured for a user, e-mails received through the primary or alternative e-mail addresses are forwarded to them. A copy of the messages can optionally be stored in the user's mailbox. Forward e-mail addresses do not have to be unique and their domain part does not have to be registered in Univention Management Console.

### Note

E-mail addresses can consist of the following characters: letters a-z, figures 0-9, dots, hyphens and underscores. The address has to begin with a letter and must include an @ character. At least one mail domain must be registered for to be able to assign e-mail addresses (see Section 14.3.1).

E-mail addresses are managed in the UMC module *Users*. The **primary e-mail address** is entered in the **General** tab in the **User account** submenu. **Alternative e-mail addresses** can be entered under **Advanced settings -> Mail**.

### Note

Once the user account is properly configured authentication to the mail stack is possible (`IMAP`/`POP3`/`SMTP`). Please keep in mind that after disabling the account or changing the password the login to the mail stack is still possible for 5min due to the authentication cache of the mail stack. To invalidate the authentication cache run

```
doveadm auth cache flush
```

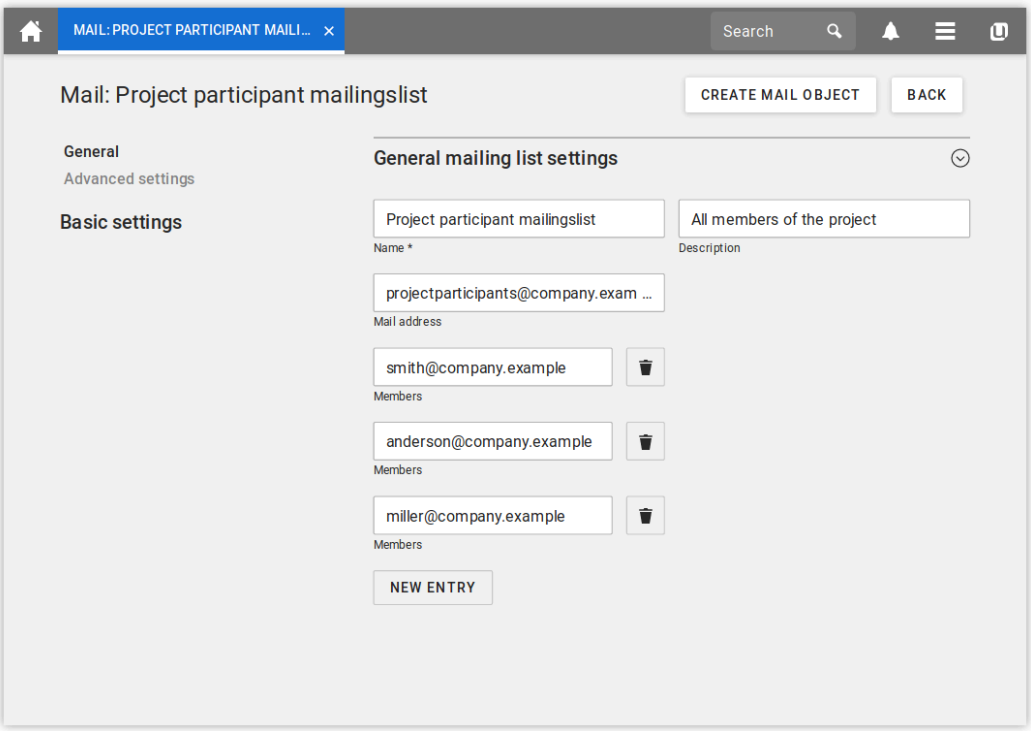
on the mail server. The expiration time of the authentication cache can be configured on the mail server with the Univention Configuration Registry variable `mail/dovecot/auth/cache_ttl` and `mail/dovecot/auth/cache_negative_ttl`.

## 14.3.3. Management of mailing lists

Feedback

Mailing lists are used to exchange e-mails in closed groups. Each mailing list has its own e-mail address. If an e-mail is sent to this address, it is received by all the members of the mailing list.

**Figure 14.1. Creating a mailing list**




Mail domains are managed in the UMC module *Mail* with the **Mailing list** object type. A name of your choice can be entered for the mailing list under **Name**; the entry of a **Description** is optional. The e-mail address of the mailing list should be entered as the **Mail address**. The domain part of the address needs to be the same as one of the managed mail domains. As many addresses as necessary can be entered under **Members**. In contrast to mail groups (see Section 14.3.4), external e-mail addresses can also be added here. The mailing list is available immediately after its creation.

In the default settings, everyone can write to the mailing list. To prevent misuse, there is the possibility of restricting the circle of people who can send mails. To do so, the Univention Configuration Registry variable `mail/postfix/policy/listfilter` on the mail server must be set to `yes` and Postfix restarted. **Users that are allowed to send e-mails to the list** and **Groups that are allowed to send e-mails to the list** can be specified under **Advanced settings**. If a field is set here, only authorized users/groups are allowed to send mails.

## 14.3.4. Management of mail groups

Feedback

There is the possibility of creating a mail group: This is where an e-mail address is assigned to a group of users. E-mails to this address are delivered to the primary e-mail address of each of the group members.

Mail groups are managed in the UMC module *Groups* (see Chapter 7).

The e-mail address of the mail group is specified in the **mail address** input field under **Advanced settings**. The domain part of the address must be the same as one of the managed mail domains.

In the default settings, everyone can write to the mail group. To prevent misuse, there is the possibility of restricting the circle of people who can send mails. To do so, the Univention Configuration Registry variable `mail/postfix/policy/listfilter` on the mail server must be set to `yes` and Postfix restarted.

**Users that are allowed to send e-mails to the group** and **Groups that are allowed to send e-mails to the group** can be specified under **Advanced settings**. If a field is set here, only authorized users/groups are allowed to send mails.

## 14.3.5. Management of shared IMAP folders

Feedback

Shared e-mail access forms the basis for cooperation in many work groups. In UCS, users can easily create folders in their own mailboxes and assign permissions so that other users may read e-mails in these folders or save additional e-mails in them.

Alternatively, individual IMAP folders can be shared for users or user groups. This type of folder is described as a shared IMAP folder. Shared IMAP folders are managed in the UMC module *Mail* with the **Mail folder (IMAP)** object type.

Shared folders cannot be renamed, therefore the Univention Configuration Registry variable `mail/dovecot/mailbox/rename` is not taken into account. When a shared folder is deleted in the UMC module *Mail*, it is only deleted from the hard disk, if `mail/dovecot/mailbox/delete` is set to `yes`. The default value is `no`.

**Figure 14.2. Creating a shared IMAP folder**

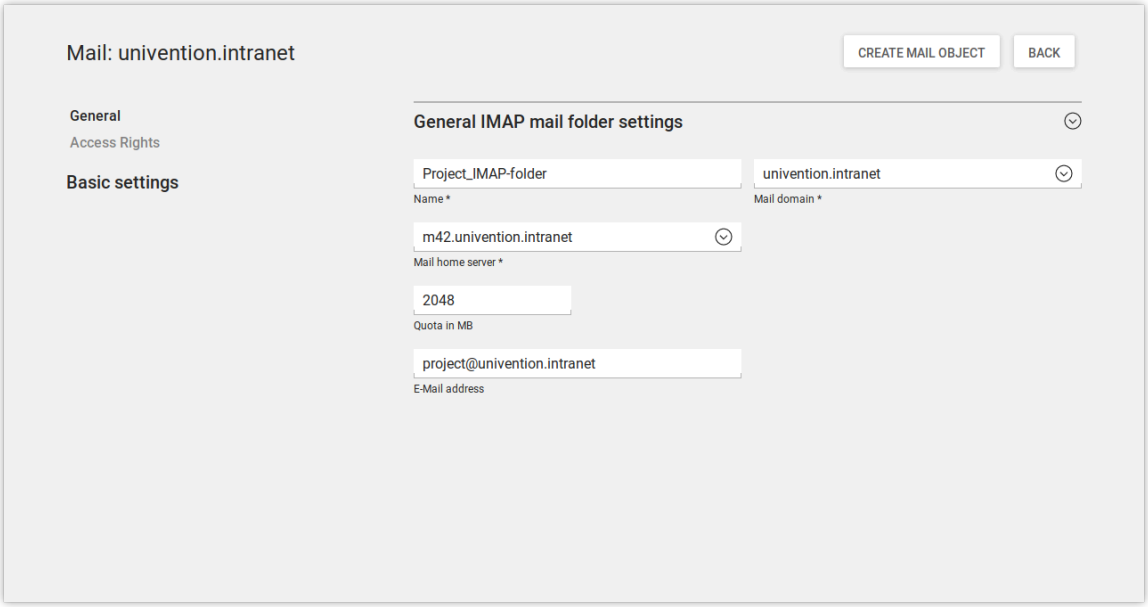



**Table 14.1. 'General' tab**

| Attribute | Description |
|---|---|
| Name (*) | The name under which the IMAP folder is available in the e-mail clients. The name displayed in the IMAP client differs depending on if an e-mail address is configured (see row "E-Mail address" below) or not. If no e-mail address is configured, the IMAP folder will be displayed in the client as `name@domain/INBOX`. If an e-mail address is configured, it will be `shared/name@domain`. |
| Mail domain (*) | Every shared IMAP folder is assigned to a mail domain. The management of the domains is documented in the Section 14.3.1. |
| Mail home server (*) | An IMAP folder is assigned to a mail home server. Further information can be found in Section 14.8.10. |

| Attribute | Description |
|---|---|
| Quota in MB | This setting can be used to set the maximum total size of all e-mails in this folder. |
| E-Mail address | An e-mail address can be entered here via which e-mails can be sent directly to the IMAP folder. If no address is set here, it is only possible to write in the folder from e-mail clients.<br><br>The domain part of the e-mail address must be registered in Univention Management Console (see Section 14.3.1). |

**Table 14.2. 'Access rights' tab**

| Attribute | Description |
|---|---|
| Name (*) | Access permissions based on users or groups can be entered here. Users are entered with their user name; the groups saved in Univention Management Console can be used as groups.<br><br>The access permissions have the following consequences for individual users or members of the specified group:<br><br>No access<br><br>No access is possible. The folder is not displayed in the folder list.<br><br>Read<br><br>The user may only perform read access to existing entries.<br><br>Append<br><br>Existing entries may not be edited; only new entries may be created.<br><br>Write<br><br>New entries may be created in this directory; existing entries may be edited or deleted.<br><br>Post<br><br>Sending an e-mail to this directory as a recipient is permitted. This function is not supported by all the clients.<br><br>All<br><br>Encompasses all permissions of *write* and also allows the changing of access permissions. |

## 14.3.6. Mail quota

Feedback

The size of the users' mailboxes can be restricted via the mail quota. When this is attained, no further e-mails can be accepted for the mailbox by the mail server until the user deletes old mails from her account.

The limit is specified in megabytes in the **Mail quota** field under **Advanced settings -> Mail**. The default value is `0` and means that no limit is set. The multi edit mode of Univention Management Console can be used to assign a quota to multiple users at one time, see Section 4.4.3.3.

The user can be warned once a specified portion of the mailbox is attained and then receives a message that his available storage space is almost full. The administrator can enter the threshold in percent and the messages subject and text:

- The threshold for when the warning message should be issued can be configured in the Univention Configuration Registry variable `mail/dovecot/quota/warning/text/PERCENT=TEXT`. `PERCENT` must be a number between 0 and 100 without the percent sign, TEXT will be the content of the e-mail.

  If `TEXT` contains the string `$PERCENT`, it will be replaced in the email with the value of the limit that has been exceeded.

  The value of the Univention Configuration Registry variable `mail/dovecot/quota/warning/subject` will be used for the subject of the e-mail.

- When the mail server package is installed, a subject and two warning messages are automatically configured:

  - `mail/dovecot/quota/warning/subject` is set to `Quota-Warning`
  - `mail/dovecot/quota/warning/text/80` is set to `Your mailbox has filled up to over $PERCENT%.`
  - `mail/dovecot/quota/warning/text/95` is set to `Attention: Your mailbox has already filled up to over $PERCENT%. Please delete some messages or contact the administrator.`

## 14.4. Spam detection and filtering

Feedback

Undesirable and unsolicited e-mails are designated as Spam. The software SpamAssassin and Postgrey are integrated in UCS for the automatic identification of these e-mails. SpamAssassin attempts to identify whether an e-mail is desirable or not based on heuristics concerning its origin, form and content. Postgrey is a policy server for Postfix, which implements gray listing. Grey listing is a Spam detection method which denies the first delivery attempt of external mail servers. Mail servers of Spam senders most often do not perform a second delivery attempt, while legitimate servers do so. Integration occurs via the packages ***univention-spamassassin*** and ***univention-postgrey***, which are automatically set up during the installation of the mail server package.

SpamAssassin operates a point system, which uses an increasing number of points to express a high probability of the e-mail being Spam. Points are awarded according to different criteria, for example, keywords within the e-mail or incorrect encodings. In the standard configuration only mails with a size of up to 300 kilobytes are scanned, this can be adjusted using the Univention Configuration Registry variable `mail/antispam/bodysizelimit`. E-mails which are classified as Spam - because they exceed a certain number of points - are not delivered to the recipient's inbox by Dovecot, but rather in the *Spam* folder below it. The name of the folder for Spam can be configured with the Univention Configuration Registry variable `mail/dovecot/folder/Spam`. The filtering is performed by a Sieve script, which is automatically generated when the user is created.

The threshold in these scripts as of which e-mails are declared to be Spam can be configured with the Univention Configuration Registry variable `mail/antispam/requiredhits`. The presetting (`5`) generally does not need to be adjusted. However, depending on experience in the local environment, this value can also be set lower. This will, however, result in more e-mails being incorrectly designated as Spam. Changes to the threshold do not apply to existing users, but the users can change the value themselves in the Horde web client (see Section 14.10.4).

There is also the possibility of evaluating e-mails with a Bayes classifier. This compares an incoming e-mail with statistical data already gathered from processed e-mails and uses this to adapt it's evaluation to the user's e-mail. The Bayes classification is controlled by the user himself, whereby e-mails not identified as Spam by

the system can be placed in the *Spam* subfolder by the user and a selection of legitimate e-mails copied into the *Ham* (`mail/dovecot/folder/ham`) subfolder. This folder is evaluated daily and data which have not yet been collected or were previously classified incorrectly are collected in a shared database. This evaluation is activated in the default setting and can be configured with the Univention Configuration Registry variable `mail/antispam/learndaily`.

The Spam filtering can be deactivated by setting the Univention Configuration Registry variable `mail/antivir/spam` to `no`. When modifying Univention Configuration Registry variables concerning Spam detection, the AMaViS service and Postfix must be restarted subsequently.

## 14.5. Identification of viruses and malware

Feedback

The UCS mail services include virus and malware detection via the ***univention-antivir-mail*** package, which is automatically set up during the set up of the mail server package. The virus scan can be deactivated with the Univention Configuration Registry variable `mail/antivir`.

All incoming and outgoing e-mails are scanned for viruses. If the scanner recognizes a virus, the e-mail is sent to quarantine. That means that the e-mail is stored on the server where it is not accessible to the user. The original recipient receives a message per e-mail stating that this measure has been taken. If necessary, the administrator can restore or delete this from the `/var/lib/amavis/virusmails/` directory. Automatic deletion is not performed.

The AMaViSd-new software serves as an interface between the mail server and different virus scanners. The free virus scanner ClamAV is included in the package and enters operation immediately after installation. The signatures required for virus identification are procured and updated automatically and free of charge by the Freshclam service.

Alternatively or in addition, other virus scanners can also be integrated in AMaViS. Postfix and AMaViS need to be restarted following changes to the AMaViS or ClamAV configuration.

## 14.6. Identification of Spam sources with *DNS-based Blackhole Lists* (DNSBL)

Feedback

Another means of combating Spam is to use a *DNS-based Blackhole List* (DNSBL) or *Real-time Blackhole List* (RBL). DNSBLs are lists of IP addresses that the operator believes to be (potential) sources of Spam. The lists are checked by DNS. If the IP of the sending e-mail server is known to the DNS server, the message is rejected. The IP address is checked quickly and in a comparatively resource-friendly manner. The check is performed *before* the message is accepted. The extensive checking of the content with SpamAssassin and anti-virus software is only performed once it has been received. Postfix has integrated support for DNSBLs (http://www.postfix.org/postconf.5.html#reject_rbl_client).

DNSBLs from various projects and companies are available on the Internet. Please refer to the corresponding websites for further information on conditions and prices.

The Univention Configuration Registry variable `mail/postfix/smtpd/restrictions/recipient/`*`SEQUENCE=RULE`* must be set to be able to use DNSBLs with Postfix. It can be used to configure recipient restrictions via the Postfix option *smtpd_recipient_restrictions* (see http://www.postfix.org/postconf.5.html#smtpd_recipient_restrictions). The sequential number is used to sort multiple rules alphanumerically, which can be used to influences the ordering.

### Tip

Existing *smtpd_recipient_restrictions* regulations can be listed as follows:

```
ucr search --brief mail/postfix/smtpd/restrictions/recipient
```

In an unmodified Univention Corporate Server Postfix installation, the DNSBL should be added to the end of the *smtpd_recipient_restrictions* rules. For example:

```
ucr set mail/postfix/smtpd/restrictions/recipient/80="reject_rbl_client
 ix.dnsbl.manitu.net"
```

# 14.7. Integration of Fetchmail for retrieving mail from external mailboxes

Feedback

Usually, the UCS mail service accepts mails for the users of the UCS domain directly via SMTP. UCS also offers optional integration of the software Fetchmail for fetching emails from external POP3 or IMAP mailboxes.

Fetchmail can be installed via the Univention App Center; simply select the **Fetchmail** application and then click on **Install**.

Once the installation is finished, there are additional input fields in the **Advanced settings -> Remote mail retrieval** tab of the user administration which can be used to configure the collection of mails from an external server. The mails are delivered to the inboxes of the respective users (the primary e-mail address must be configured for that).

The mail is fetched every twenty minutes once at least one e-mail address is configured for mail retrieval. After the initial configuration of a user Fetchmail needs to be started in the UMC module **System services**. In that module the fetching can also be disabled (alternatively by setting the Univention Configuration Registry variable `fetchmail/autostart` to `false`).

**Table 14.3. 'Remote mail retrieval' tab'**

| Attribute | Description |
|---|---|
| Username | The user name to be provided to the mail server for fetching mail. |
| Password | The password to be used for fetching mail. |
| Protocol | The mail can be fetched via the IMAP or POP3 protocols. |
| Remote mail server | The name of the mail server from which the e-mails are to be fetched. |
| Encrypt connection (SSL/TLS) | If this option is enabled, the mail is fetched in an encrypted form (when this is supported by the mail server). |
| Keep mails on the server | In the default settings, the fetched mails are deleted from the server following the transfer. If this option is enabled, it can be suppressed. |

# 14.8. Configuration of the mail server

Feedback

## 14.8.1. Configuration of a relay host for sending the e-mails

Feedback

In the default setting, Postfix creates a direct SMTP connection to the mail server responsible for the domain when an e-mail is sent to a non-local address. This server is determined by querying the MX record in the DNS.

Alternatively, a mail relay server can also be used, i.e., a server which receives the mails and takes over their further sending. This type of mail relay server can be provided by a superordinate corporate headquarters or the Internet provider, for example. To set a relay host, it must be entered as a fully qualified domain name (FQDN) in the Univention Configuration Registry variable `mail/relayhost`.

If authentication is necessary on the relay host for sending, the Univention Configuration Registry variable `mail/relayauth` must be set to `yes` and the `/etc/postfix/smtp_auth` file edited. The relay host, user name and password must be saved in this file in one line.

```
FQDN-Relayhostusername:password
```

The command

```
postmap /etc/postfix/smtp_auth
```

must then be executed for this file to adopt the changes via Postfix.

**Note**

To ensure an encrypted connection while using a relay host, the Postfix option `smtp_tls_security_level=encrypt` has to be set. Univention Corporate Server will set this option automatically, if `mail/relayhost` is set and `mail/relayauth` is set to `yes` and `mail/postfix/tls/client/level` is not set to `none`.

## 14.8.2. Configuration of the maximum mail size

Feedback

The Univention Configuration Registry variable `mail/messagesizelimit` can be used to set the maximum size in bytes for incoming and outgoing e-mails. Postfix must be restarted after modifying the setting. The preset maximum size is 10240000 bytes. If the value is configured to `0` the limit is effectively removed. Please note that e-mail attachments are enlarged by approximately a third due to the base64 encoding.

If Horde (see Section 14.10) is used, the Univention Configuration Registry variables `php/limit/filesize` and `php/limit/postsize` must also be adjusted. The maximum size in megabytes must be entered as the value in both variables. Then the Apache web server has to be restarted.

## 14.8.3. Configuration of a blind carbon copy for mail archiving solutions

Feedback

If the Univention Configuration Registry variable `mail/archivefolder` is set to an e-mail address, Postfix sends a blind carbon copy of all incoming and outgoing e-mails to this address. This results in an archiving of all e-mails. The e-mail address must already exist. It can be either one already registered in Univention Corporate Server as the e-mail address of a user, or an e-mail account with an external e-mail service. As standard the variable is not set.

Postfix must then be restarted.

## 14.8.4. Configuration of soft bounces

Feedback

If a number of error situations (e.g., for non-existent users) the result may be a mail bounce, i.e., the mail cannot be delivered and is returned to the sender. When Univention Configuration Registry variable `mail/postfix/softbounce` is set to `yes` e-mails are never returned after a bounce, but instead are held in the queue. This setting is particularly useful during configuration work on the mail server.

## 14.8.5. Configuration of SMTP ports

Feedback

On a Univention Corporate Server mail server Postfix is configured to listen for connections on three ports:

- Port 25 (`SMTP`) should be used by other mail servers only. By default authentication is disabled. If submission of emails from users is desired on port 25, authentication can be enabled by setting the Univention Configuration Registry variable `mail/postfix/mastercf/options/smtp/smtpd_sasl_auth_enable=yes`.

- Port 465 (`SMTPS`) allows authentication and email submission through a SSL encrypted connection. `SMTPS` has been declared deprecated in favor of port 587 but is kept enabled for legacy clients.
- Port 587 (`Submission`) allows authentication and email submission through a TLS encrypted connection. The use of STARTTLS is enforced.

The Submission port should be preferred by email clients. The use of the ports 25 and 465 for email submission is deprecated.

## 14.8.6. Configuration of additional checks by postscreen

Feedback

When using a mail server that is directly accessible from the Internet, there is always a risk that Spam sender, Spam bots or broken mail servers are continually trying to deliver unwanted emails (for example Spam) to the UCS system.

To reduce the load of the mail server for such cases, Postfix brings its own service with the name postscreen, which is put in front of Postfix and accepts incoming SMTP connections. On these incoming SMTP connections, some lightweight tests are first performed. If the result is positive, the respective connection is passed on to Postfix. Otherwise the SMTP connection is terminated and thus the incoming mail is rejected before being in the area of responsibility of the UCS mail server.

By default, postscreen is not active. By setting the Univention Configuration Registry variable `mail/postfix/postscreen/enabled` to the value `yes`, postscreen can be activated.

Various UCR variables with the prefix `mail/postfix/postscreen/` can be used to configure postscreen. A list of all relevant UCR variables including descriptions can be retrieved e.g. on command line via the command `ucr search --verbose mail/postfix/postscreen/`.

### Note

After each change of a UCR variable for postscreen the configuration of Postfix and postscreen should be reloaded. This can be triggered e.g. via the command `service postfix reload`.

## 14.8.7. Custom Postfix configuration

Feedback

It is possible to modify the Postfix configuration, that resides within the file `/etc/postfix/main.cf`, beyond the variables that can be set with Univention Configuration Registry variable.

If the file `/etc/postfix/main.cf.local` exists, its content will be appended to the file `main.cf`. To transfer changes of `main.cf.local` to `main.cf`, the following command must be executed:

```
ucr commit /etc/postfix/main.cf
```

For the Postfix service to accept the changes, it must be reloaded:

```
service postfix reload
```

If a Postfix variable that has previously been set in `main.cf` is set again in `main.cf.local`, Postfix will issue a warning to the log file `/var/log/mail.log`.

### Note

If Postfix' behavior is not as expected, first remove configuration settings made by `main.cf.local`. Rename the file or comment out its content. Next run the two commands above. The configuration will then return to UCS defaults.

## 14.8.8. Configuring the Alias-expansion-limit

Feedback

When sending a mail to a group including other nested groups, the mail may not be accepted/delivered. This is caused by Postfix trying to expand the number of the primary recipients via Virtual Alias Expansion. This number is limited to 1000 users by default which might be too low.

To adjust the number to (for instance) 5000 users, the following line can be added or edited in `/etc/postfix/main.cf.local`:

```
virtual_alias_expansion_limit = 5000
```

Afterwards Postfix needs to be restarted:

```
service postfix restart
```

## 14.8.9. Handling of mailboxes during e-mail changes and the deletion of user accounts

Feedback

A user's mailbox is linked to the primary e-mail address and not to the user name. The Univention Configuration Registry variable `mail/dovecot/mailbox/rename` can be used to configure the reaction when the primary e-mail address is changed:

- If the variable is set to `yes`, the name of the user's IMAP mailbox is changed. This is the standard setting since UCS 3.0.
- If the setting is `no`, it will not be possible to read previous e-mails any more once the user's primary e-mail address is changed! If another user is assigned a previously used primary e-mail address, she receives access to the old IMAP structure of this mailbox.

The Univention Configuration Registry variable `mail/dovecot/mailbox/delete` can be used to configure, whether the IMAP mailbox is also deleted. The value `yes` activates the removal of the corresponding IMAP mailbox if one of the following actions is performed:

- deletion of the user account
- removal of the primary e-mail address from the user account
- changing the user's mail home server to a different system

With default settings (`no`) the mailboxes are kept if one of the actions above is performed.

The combination of the two variables creates four possible outcomes when the e-mail address is changed:

**Table 14.4. Renaming of e-mail addresses**

| `mail/dovecot/mailbox/…` | **Meaning** |
|---|---|
| `rename=yes` and `delete=no` (Default) | The existing mailbox will be renamed. E-mails will be preserved and will be accessible at the new address. |
| `rename=yes` and `delete=yes` | The existing mailbox will be renamed. E-mails will be preserved and will be accessible at the new address. |
| `rename=no` and `delete=no` | A new, empty mailbox will be created. The old one will be preserved on disk with the old name and will thus not be accessible to users. |

| mail/dovecot/mailbox/… | Meaning |
|---|---|
| rename=no and delete=yes | A new, empty mailbox will be created. The old one will be deleted from the hard disk. |

## 14.8.10. Distribution of an installation on several mail servers

Feedback

The UCS mail system offers the possibility of distributing users across several mail servers. To this end, each user is assigned a so-called mail home server on which the user's mail data are stored. When delivering an e-mail, the responsible home server is automatically determined from the LDAP directory.

It must be observed that global IMAP folders (see Section 14.3.5) are assigned to a mail home server.

If the mail home server is changes for a user, the user's mail data is *not* moved to the server automatically.

## 14.8.11. Mail storage on NFS

Feedback

Dovecot supports storing e-mails and index files on cluster filesystems and on NFS. Some settings are necessary to prevent data loss in certain situations.

The following settings assume that mailboxes are not accessed simultaneously by multiple servers. This is the case if for each user exactly one mail home server has been configured.

- `mail/dovecot/process/mmap_disable = yes`
- `mail/dovecot/process/dotlock_use_excl = yes`
- `mail/dovecot/process/mail_fsync = always`

To achieve higher performance, index files can be kept on the local servers disk, instead of storing them together with the messages on NFS. The index files can then be found at `/var/lib/dovecot/index/`. To activate this option, set Univention Configuration Registry variable `mail/dovecot/location/separate_index = yes`.

With the above settings the mail server should work without problems on NFS. There are however a lot of different client and server systems in service. In case you encounter problems, here are some notes that might help:

- If NFSv2 is in use (not the case if the NFS server is a Univention Corporate Server), please set `mail/dovecot/process/dotlock_use_excl = no`.
- If lockd is not in use (not the case on Univention Corporate Server systems) or if even with lockd in use locking error are encountered, set `mail/dovecot/process/lock_method = dotlock`. This does lower the performance, but solves most locking related errors.
- Dovecot flushes NFS caches when needed if you set `mail/dovecot/process/mail_nfs_storage = yes`, but unfortunately this doesn't work 100%, so you can get random errors. The same holds for flushing NFS caches after writing index files with `mail/dovecot/process/mail_nfs_index = yes`.
- The Dovecot documentation has more information on the topic: [dovecot-wiki-clusterfs] [dovecot-wiki-nfs]

## 14.8.12. Connection limits

Feedback

In a default Univention Corporate Server configuration Dovecot allows 400 concurrent IMAP and POP3 connections each. That is enough to serve at least 100 concurrently logged in IMAP users, possibly a lot

more. How many IMAP connections are opened by a user depends on the clients they use. Webmail opens just a few short lived connections. Desktop clients keep multiple connections open over a long period of time. Mobile clients keep just a few connections open over a long period of time. But they tend to never close them, unnecessarily wasting resources. The limits exist mainly to resist denial of service attacks that open a lot of connections and create lots of processes.

To list the open connections, run:

```
doveadm who
```

To display the total amount of open connections, run:

```
doveadm who -1 | wc -l
```

The Univention Configuration Registry variables `mail/dovecot/limits/*` can be set to modify the limits. The process of adapting those variables is only semi automatic, because of their complex interaction. For the meaning of each variable refer to the Dovecot documentation: [dovecot-wiki-services]

Dovecot uses separate processes for login and to access emails. The limits for these can be configured separately. The maximum number of concurrent connections to a service and the maximum number of processes for a service is also configured separately. Setting `mail/dovecot/limits/default_client_limit = 3000` changes the limit for the maximum number of concurrent connections to the IMAP and POP3 services but does not change the maximum number of processes allowed to run. With the Univention Corporate Server default settings Dovecot runs in "High-security mode": each connection is served by a separate process. The default is to allow only 400 processes, so only 400 connections can be made.

To allow 3000 clients to connect to their emails, another Univention Configuration Registry variable has to be set:

```
ucr set mail/dovecot/limits/default_client_limit=3000
ucr set mail/dovecot/limits/default_process_limit=3000
doveadm reload
```

Reading `/var/log/dovecot.info` reveals a warning:

```
config: Warning: service auth { client_limit=2000 } is lower than
 required under max. load (15000)
config: Warning: service anvil { client_limit=1603 } is lower than
 required under max. load (12003)
```

The services `auth` (responsible for login and SSL connections) and `anvil` (responsible for statistics collection) are set to their default limits. Although 3000 POP3 and IMAP connections and processes are allowed, the connection limit for the login service is too low. Leaving it like this will lead to failed logins.

The values are so high, because `default_client_limit` and `default_process_limit` do not only lift limits for IMAP and POP3, but also for other services like `lmtp` and `managesieve-login`. Those services can now start more processes that have to be monitored and can theoretically make more authentication requests. This increases the number of possible concurrent connections to the `auth` and `anvil` services.

The values have to be adapted, using the numbers from the log file:

```
ucr set mail/dovecot/limits/auth/client_limit=15000
ucr set mail/dovecot/limits/anvil/client_limit=12003
doveadm reload
```

Another warning appears in `/var/log/dovecot.info`:

```
master: Warning: fd limit (ulimit -n) is lower than required under max.
 load (2000 < 15000),…
 because of service auth { client_limit }
```

The Linux kernel controlled setting `ulimit` setting (limit on the number of files/connections a process is allowed to open) is changed only when the Dovecot service is restarted:

```
invoke-rc.d dovecot restart
```

No more warnings are written to the log file and both IMAP and POP3 servers now accept 3000 client connections each.

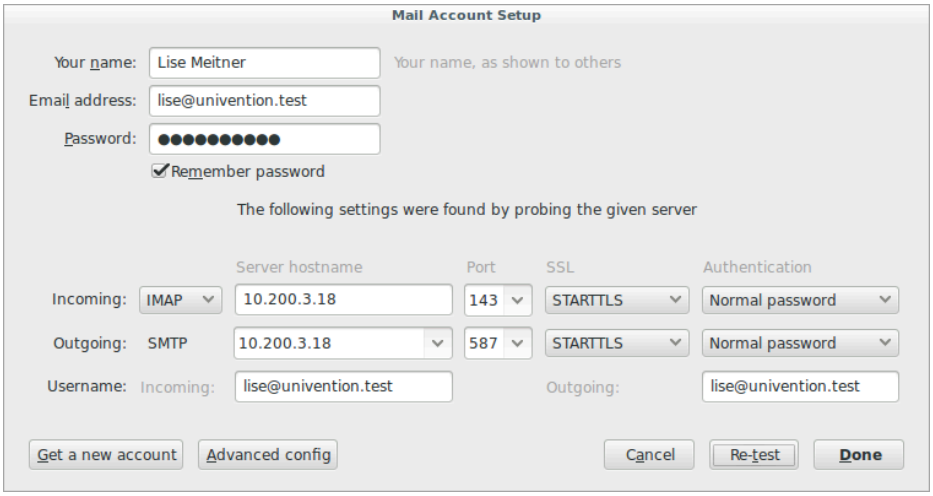
Univention Corporate Server configures Dovecot to run in "High-security mode" by default. For installations with 10.000s of users, Dovecot offers the "High-performance mode". The performance guide has further details on how to configure it: [ucs-performance-guide].

## 14.9. Configuration of mail clients for the mail server

Feedback

The use of IMAP is recommended for using a mail client with the UCS mail server. STARTTLS is used to switch to a TLS-secured connection after an initial negotiation phase when using SMTP (for sending mail) and IMAP (for receiving/synchronizing mail). *Password (plain text)* should be used in combination with *STARTTLS* as the authentication method. The method may have a different name depending on the mail client. The following screenshot shows the setup of Mozilla Thunderbird as an example.

**Figure 14.3. Setup of Mozilla Thunderbird**




## 14.10. Webmail and administration of e-mail filters with Horde

Feedback

UCS integrates a number of applications from the Horde framework for web access to e-mails and web-based administration of server-side e-mail filter rules based on Sieve. Horde can be installed via the Univention App Center (see Section 5.3).

### 14.10.1. Login and overview

Feedback

The Horde login mask is linked on the system overview page (see Section 4.2) under **Horde web client** and can be opened directly at `http://SERVERNAME/horde/`.

**Figure 14.4. Login on Horde**

Either the UCS user name or the primary e-mail address can be used as the user name. The webmail interface can be used in a number of display modes. The preferred version can be selected under **Mode**. We recommend the use of the dynamic interface for standard workstations. The remaining documentation refers to this version. Selecting the **Language** has no effect in many web browsers, as the browser's preferred language settings take precedence.

In the top toolbar there are a number or menu points (e.g., **Mail** and **Address Book**), which can be used to switch between the individual modules.

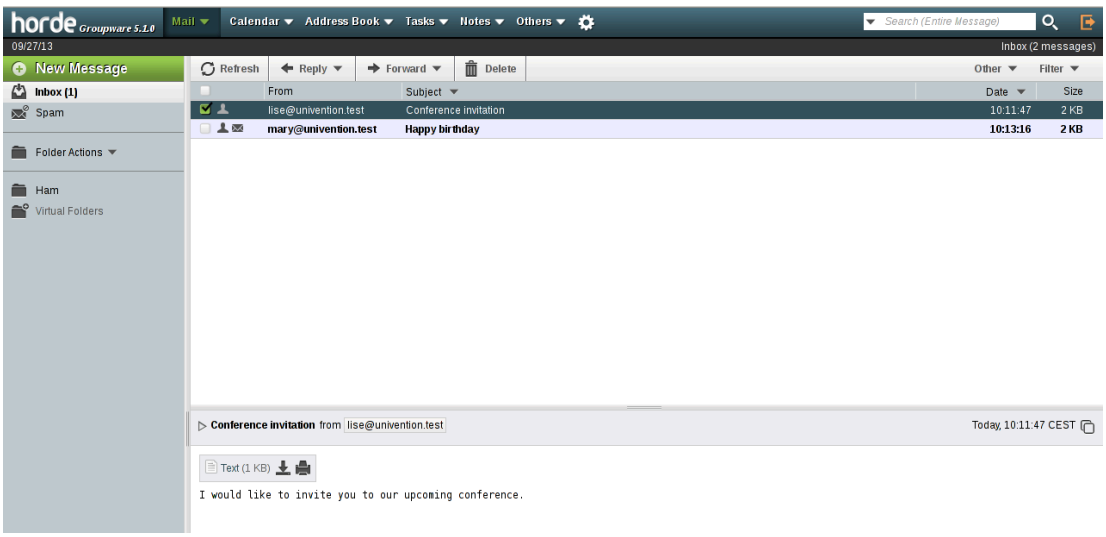
The user can personalize Horde by clicking the cog symbol.

## 14.10.2. Web-based mail access

Feedback

Horde offers all the standard functions of an e-mail client such as the sending, forwarding and deletion of e-mails. E-mails can be sorted in folders and are stored in **Inbox** as standard. A *Sent* folder is created automatically the first time an e-mail is sent.

**Figure 14.5. Web mail (Inbox)**



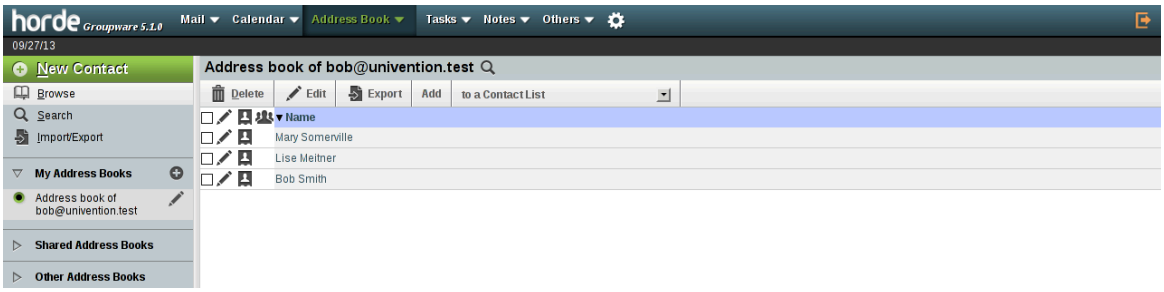
Horde differentiates between two types of deletion: An e-mail deleted with **Delete** is initially moved to the *Trash* folder. From there, it can be moved into any other folder as long as the trash can has not been emptied with **Empty**.

## 14.10.3. Address book

Feedback

This module is used to administrate e-mail addresses and additional contact information. The information compiled here are saved in Horde's own SQL database.

**Figure 14.6. Address book for webmail**




Contact information found using the simple or advanced search can then be copied into individual address books and edited there. New contacts can be entered via the **New Contact** menu item. Personal address books can also be created via **My Address Books**.

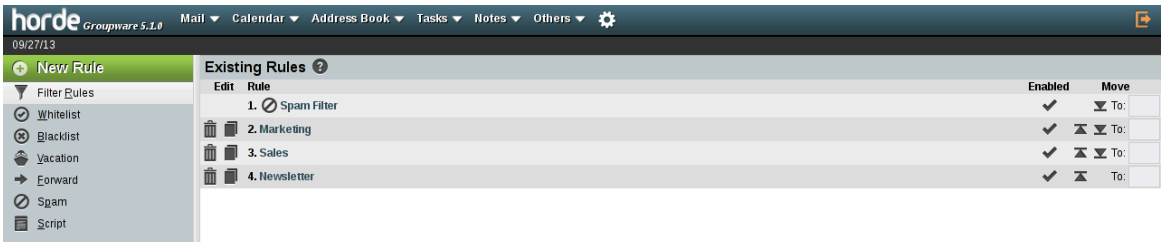
The **Browse** menu item can be used to display the contents of address books. The lists can be sorted alphabetically by clicking on the preferred column title (surname, first name, etc.). Clicking on the magnifying glass in the header of the respective address book (directly next to the name of the address book) opens a search field that can be used easily to search within the open address book. Individual addresses in a list can be marked with an X for subsequent use, i.e., to export them as a file in a certain file format or to copy them into another address book.

## 14.10.4. E-mail filters

Feedback

Dovecot supports server-side filter scripts written in an individual script language called Sieve. The filter module allows the generation of these filter scripts. They apply generally and thus also apply for users accessing their inboxes via a standard mail client.

**Figure 14.7. Filter management in Horde**




Filters can be edited and expanded under **Mail -> Filters**. The filters are applied to incoming e-mails in the consecutively numbered order. Their position can be altered either using the arrows to the right or by entering a number in the `Move` column directly. Individual filter rules can be switched on and off in the `Enabled` column.

The **Spam** filter can be used user-specifically to set which Spam threshold should apply. The specified **Spam Level** is the SpamAssassin threshold. An e-mail which returns this value will be sent to the specified folder.

A **Vacation** filter can be used to specify a period in which incoming e-mails are automatically replied to with an answer e-mail by the mail server. The text and subject of the e-mail can be selected as required.

**New Rule** can be used to create new rules, e.g., for the automatic sorting of incoming mails into topic-specific mail folders.

Clicking on **Script** displays the source text of the generated Sieve script.

# Chapter 15. Infrastructure monitoring

15.1. Introduction ..... 271
15.1.1. UCS Dashboard ..... 271
15.1.1.1. Introduction and structure ..... 271
15.1.1.2. Installation ..... 271
15.1.1.3. Usage ..... 272
15.2. Nagios ..... 273
15.2.1. Introduction and structure ..... 273
15.2.2. Installation ..... 275
15.2.2.1. Preconfigured Nagios checks ..... 275
15.2.3. Configuration of the Nagios monitoring ..... 277
15.2.3.1. Configuration of a Nagios service ..... 277
15.2.3.2. Configuration of a monitoring time period ..... 280
15.2.3.3. Assignment of Nagios checks to computers ..... 280
15.2.3.4. Integration of additional Nagios plugin configurations ..... 282
15.2.4. Querying the system status via the Nagios web interface ..... 282
15.2.5. Integration of additional plugins ..... 282

## 15.1. Introduction

Feedback

UCS offers two different solutions for infrastructure monitoring. On the one hand the UCS Dashboard helps administrators to quickly read the state of domains and individual servers. On the other hand, with Nagios it is possible to continuously check computers and services in the background and proactively trigger a notification if a warning level is reached.

The following sections describe the two different solutions.

### 15.1.1. UCS Dashboard

Feedback

#### 15.1.1.1. Introduction and structure

Feedback

The UCS Dashboard app allows administrators to view the state of the domain and individual servers can be read quickly and clearly on so-called dashboards. The dashboards are accessible via a Web browser, access a database in the background, and deliver continuously updated reports on specific aspects of the domain or server.

Prerequisite for using the *UCS Dashboard* app is a valid subscription. More information can be found on the website https://www.univention.com/products/prices-and-subscriptions/.

#### 15.1.1.2. Installation

Feedback

The UCS Dashboard consists of three parts:

- The UCS Dashboard app for the visualization of data from the central Database. This component is based on the software component Grafana.
- The UCS Dashboard Database app, a time series database for storing of the metrics. This database is based on the software Prometheus.
- The UCS Dashboard Client app for deploying the metrics of server systems. This is based on the Prometheus Node Exporter.

The app *UCS Dashboard* can be installed from the Univention App Center on a server in the domain. Currently, the installation is only possible on the system roles Domain controller master, backup or slave. The apps *UCS Dashboard Database* and *UCS Dashboard Client* are automatically installed on the same system.

The app *UCS Dashboard Client* should be installed on every UCS system. Only then will the system data be displayed on the dashboard.

### 15.1.1.3. Usage

Feedback

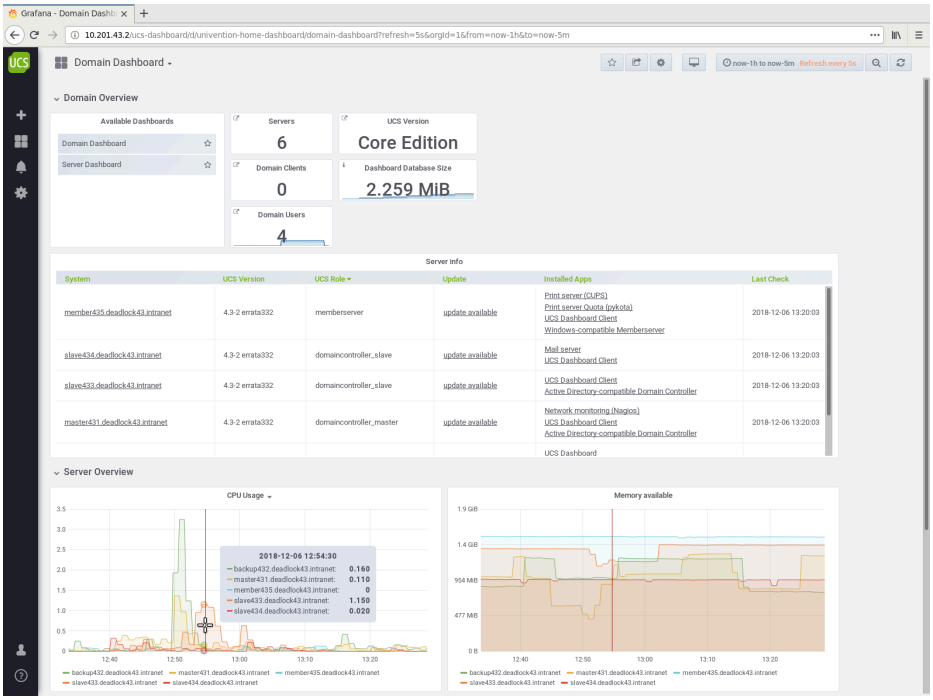
After the installation, the UCS Dashboard is linked in the portal. Alternatively, it can be accessed directly via `https://SERVERNAME-OR-IP/ucs-dashboard/`.

In the default setting, access is only granted to users of the group `Domain Admins` (e.g. the user Administrator).

#### 15.1.1.3.1. Domain dashboard

Feedback

**Figure 15.1. Domain dashboard**




After the login, the Domain dashboard is opened by default. On this dashboard, general information about the domain is displayed, such as how many servers and how many users exist in the environment.

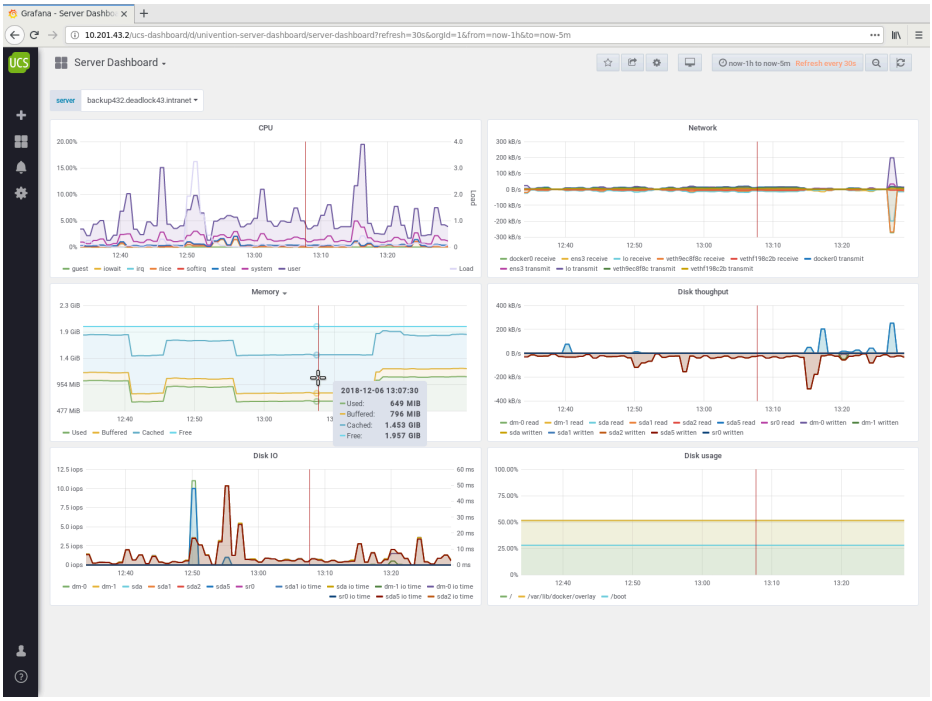
Furthermore, the UCS systems are listed on the dashboard, in a tabular overview, including further information, such as the server role, the installed apps or whether an update is available or not.

In addition, the CPU usage, memory usage, free hard disk space and the status of the LDAP replication are displayed. In this graphics all servers are displayed together.

#### 15.1.1.3.2. Server dashboard

Feedback

**Figure 15.2. Server dashboard**




By default, the server dashboard is also configured. On this dashboard, detailed information about individual server systems are shown, such as the CPU- or memory usage or network throughput.

The servers can be selected in the dropdown **server**. Then the graphics show the details about the selected server.

#### 15.1.1.3.3. Own dashboards

Feedback

The two included dashboards *domain dashboard* and *server dashboard* can't be changed, because they are updated by Univention with updates.

Instead, you can create your own dashboards. On these dashboards you can then either add already existing elements or new elements can be created. All you need to do is click on the plus sign on the left side. A new dashboard will be created which can be filled with elements.

# 15.2. Nagios

Feedback

## 15.2.1. Introduction and structure

Feedback

With the help of the Nagios software, it is possible to verify the correct function of complex IT structures from networks, computers and services continually and automatically.

Nagios has a comprehensive collection of monitoring modules, the so-called Nagios plugins. In addition to polling system indicators (e.g., CPU and memory utilization, free disk space), they also allow to test the availability and function of different services (e.g., SSH, SMTP, HTTP). Simple program steps such as the delivery of a test e-mail or the resolution of a DNS record are generally performed for the function tests. In

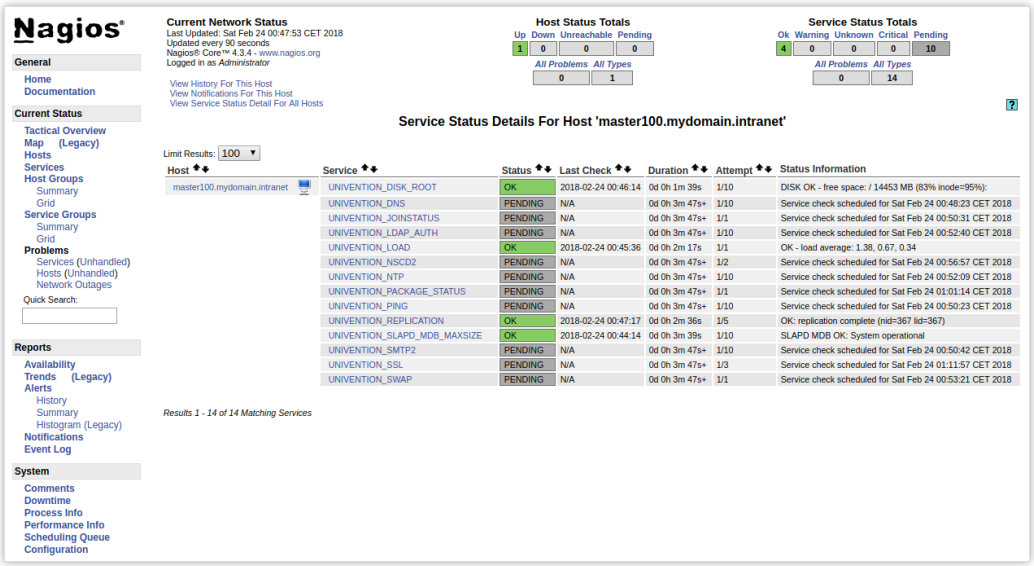
addition to the standard plugins included in Nagios, the UCS-specific plugins are also provided, with which the listener/notifier replication can be monitored, for example.

Nagios differentiates between three basic operating statuses for a service:

- *OK* is regular operation
- *CRITICAL* describes an error, e.g., a web server which cannot be reached
- *WARNING* signals the possibility of an error status occurring soon and is thus a precursor of *CRITICAL*. Example: The test for sufficient free disk space on the root partition only triggers an error as of 90% full, but a warning is given as of 75%.

When the operating status changes, a contact person specified in advance can be informed of the possible malfunction. In addition to the reactive notification in case of error, the current status can also be checked at any time continually in a web-based interface in which the status information is displayed in a compact manner.

**Figure 15.3. Nagios status webinterface**




Nagios is composed of three main components:

- The core component of a Nagios installation is the *Nagios server*, which is responsible for the collection and storage of the monitoring data.
- The actual collection of the status information is performed by *Nagios plugins*, which are run at regular intervals by the Nagios server. The information gathered is saved on the Nagios server.
- Some status information cannot be requested over the network (e.g., the query of free disk space on a hard drive partition). In this case, the NRPED service (Nagios Remote Plugin Executor Daemon) is used, which runs Nagios plugins on another computer following a request from the Nagios server and then transfers the gathered information. The NRPED is provided by the *Nagios client* component, which is preinstalled on all UCS system roles.

The Nagios configuration is performed in Univention Management Console, the Nagios configuration files are automatically generated from the information stored in the LDAP directory.

## 15.2.2. Installation

Feedback

A Nagios server can be installed from the Univention App Center with the application *Network monitoring (Nagios)*. Alternatively, the software package ***univention-nagios-server*** can be installed (subsequently `univention-run-join-scripts` must be run). Additional information can be found in Section 5.6. The Nagios server can be installed on any system role; the use of a domain controller system is recommended. The Nagios client is installed by default on all system roles.

If the Nagios server is installed on another system than the domain controller master, the Univention Configuration Registry variable `nagios/client/allowedhosts` must be set to the FQDN of the Nagios server on all systems on which the Nagios client is installed. The easiest way to do this is to implement a Univention Configuration Registry policy, see Section 8.3.4 for more information.

In addition to the standard plugins provided with the installation of the ***univention-nagios-client*** package, additional plugins can be subsequently installed with the following packages:

- ***univention-nagios-raid*** Monitoring of the software RAID status
- ***univention-nagios-smart*** Test of the S.M.A.R.T. status of hard drives
- ***univention-nagios-opsi*** Test of software distribution opsi
- ***univention-nagios-ad-connector*** Test of the AD Connector

Some of the packages are automatically set up during installation of the respective services. For example, if the UCS AD connector is set up, the monitoring plugin is included automatically.

### 15.2.2.1. Preconfigured Nagios checks

Feedback

During the installation, basic Nagios tests are set up automatically for UCS systems. The mounting of additional services in documented in the Section 15.2.3.1.

| Nagios service | Description |
|---|---|
| `UNIVENTION_PING` | tests the availability of the monitored UCS system with the command `ping`. In the default setting, an error status is attained if the response time exceeds 50 ms or 100 ms or package package losses of 20% or 40% occur. |
| `UNIVENTION_DISK_ROOT` | monitors how full the `/` partition is. An error status is raised if the remaining free space falls below 25% or 10% in the default setting. |
| `UNIVENTION_DNS` | tests the function of the local DNS server and the accessibility of the public DNS server by querying the hostname `www.univention.de`. If no DNS forwarder is defined for the UCS domain, this request fails. In this case, `www.univention.de` can be replaced with the FQDN of the domain controller master for example, in order to test the function of the name resolution. |
| `UNIVENTION_LDAP` | monitors the LDAP server running on UCS domain controller systems. |
| `UNIVENTION_LOAD` | monitors the system load. |
| `UNIVENTION_NTP` | requests the time from the NTP service on the monitored UCS system. If this deviates by more than 60 or 120 seconds, the error status is attained. |
| `UNIVENTION_SMTP` | tests the mail server. |

| Nagios service | Description |
|---|---|
| UNIVENTION_SSL | tests the remaining validity period of the UCS SSL certificates. This plugin is only suitable for master domain controller and backup domain controller systems. |
| UNIVENTION_SWAP | monitors the utilization of the swap partition. An error status is raised if the remaining free space falls below the threshold (40% or 20% in the default setting). |
| UNIVENTION_REPLICATION | monitors the status of the LDAP replication and recognizes the creation of a `failed.ldif` file and the standstill of the replication and warns of large differences between the transaction IDs. |
| UNIVENTION_NSCD | tests the availability of the name server cache daemon. If there is no NSCD process running, a CRITICAL event is triggered; if more than one process is running, a WARNING. |
| UNIVENTION_WINBIND | tests the availability of the Winbind service. If no process is running, a CRITICAL event is triggered. |
| UNIVENTION_SMBD | tests the availability of the Samba service. If no process is running, a CRITICAL event is triggered. |
| UNIVENTION_NMBD | tests the availability of the NMBD service, which is responsible for the NetBIOS service in Samba. If no process is running, a CRITICAL event is triggered. |
| UNIVENTION_JOINSTATUS | tests the join status of a system. If a system has yet to join, a CRITICAL event is triggered; if non-run join scripts are available, a WARNING event is returned. |
| UNIVENTION_KPASSWD | tests the availability of the Kerberos password service (only available on domain controller master/backup). If fewer or more than one process is running, a CRITICAL event is triggered. |
| UNIVENTION_CUPS | monitors the CUPS daemon. If there is no `cupsd` process running or the web interface on port 631 is not accessible, the CRITICAL status is returned. |
| UNIVENTION_DANSGUARDIAN | monitors the DansGuardian web filter. If no DansGuardian process is running or the DansGuardian proxy is not accessible, the CRITICAL status is returned. |
| UNIVENTION_SQUID | monitors the Squid proxy. If no squid process is running or the Squid proxy is not accessible, the CRITICAL status is returned. |
| UNIVENTION_LIBVIRTD_KVM | tests the status of a KVM virtualization server via a request to `virsh` and returns CRITICAL if the request takes longer than ten seconds. |
| UNIVENTION_LIBVIRTD_XEN | tests the status of a Xen virtualization server via a request to `virsh` and returns CRITICAL if the request takes longer than ten seconds. |
| UNIVENTION_UVMMD | tests the status of the UCS Virtual Machine Manager by requesting the available nodes. If they cannot be resolved, CRITICAL is returned. |

Default parameters have been set for the services listed above, which are customized to the requirements of most UCS installations. If the default parameters are not suitable, they can also be altered subsequently. This is documented in Section 15.2.3.1.

The following Nagios services are only available on the respective Nagios client once additional packages have been installed (see Section 15.2.2):

| Nagios service | Description |
|---|---|
| UNIVENTION_OPSI | monitors the opsi daemon. If no opsi process is running or the opsi proxy is not accessible, the CRITICAL status is returned. |
| UNIVENTION_SMART_SDA | tests the S.M.A.R.T. status of the hard drive `/dev/sda`. Corresponding Nagios services exist for the hard drives `sdb`, `sdc` and `sdd`. |
| UNIVENTION_RAID | tests the status of the software RAID via `/proc/mdadm` and returns CRITICAL is one of the hard drives in the RAID association has failed or WARNING if a recovery procedure is in progress. |
| UNIVENTION_ADCONNECTOR | Checks the status of the AD connector. If no connector process is running, CRITICAL is reported; if more than one process is running per connector instance, a WARNING is given. If rejects occur, a WARNING is given. If the AD server cannot be reached, a CRITICAL status occurs. The plugin can also be used in multi-connector instances; the name of the instance must be passed on as a parameter. |

## 15.2.3. Configuration of the Nagios monitoring

Feedback

The following settings can be performed in Univention Management Console:

- All Nagios tests that can be assigned to a computer must be registered. This is performed via *Nagios service* objects, see Section 15.2.3.1.
- The assignment on which tests should be performed on a computer and which contact persons should be informed in the case of errors is performed on the respective computer objects.
- Nagios tests can be restricted in terms of time, e.g., so that the test of the print server is only performed on weekdays from 8 a.m. to 8 p.m. This is performed via *Nagios time period* objects, see Section 15.2.3.2.

In the basic setting, there is already a large number of tests defined for each computer, e.g., a Nagios basic configuration is set up without the need for any further adjustments.

### 15.2.3.1. Configuration of a Nagios service

Feedback

A Nagios service defines the monitoring of a service. Any number of computers can be assigned to such an object so that the Nagios plugins to be used and the testing and notification parameters of a service test can be set up on the specified computers with only one entry.

Nagios services are administrated in the UMC module *Nagios* with the object type **Nagios service** (see Section 4.4). Nagios has no LDAP interface for the monitoring configuration, instead the configuration files are generated by a listener module when adding/removing/editing a Nagios service.

**Figure 15.4. Configuring a Nagios service**

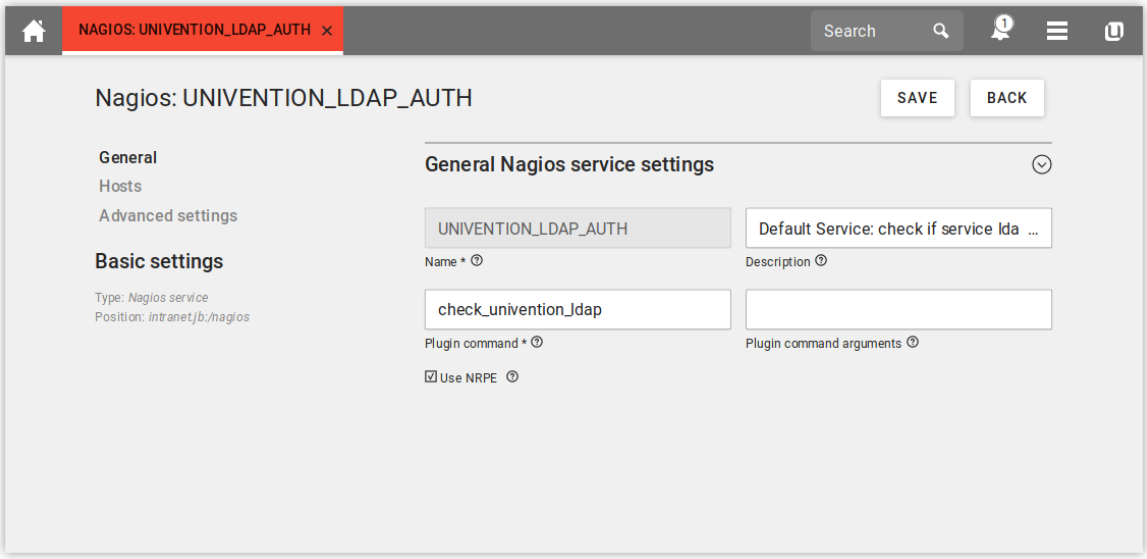



**Table 15.1. 'General' tab**

| Attribute | Description |
|---|---|
| Name | An unambiguous name for the Nagios service. |
| Description | Any description of the service. |
| Plugin command | The plugin command to be requested. Each plugin command specifies a predefined plugin execution. These are defined in the configuration files in the `/etc/nagios-plugins/config/` directory, e.g., `check_disk`. |
| Plugin command arguments | As not all parameters of the Nagios plugins are predefined in the plugin commands, it often proves necessary to enter additional parameters. The parameters specified here are separated by exclamation marks, e.g., **`20%!10%!/home`**. |
| Use NRPE | If the test of a service cannot be performed remotely (e.g., of the available drive space on the root partition), the plugin can be executed on a distant UCS system via the Nagios Remote Plugin Executor Daemon (NRPED). To do so, the ***univention-nagios-client*** package must be installed. |

**Table 15.2. 'Interval' tab (advanced settings)**

| Attribute | Description |
|---|---|
| Check interval | The check interval defines the interval of time in minutes between two service tests. |
| Retry check interval | If the last service test does not return the status *OK*, Nagios uses a different time interval for the further tests. The test frequency can be increased in this way in the case of error. If the status *OK* has not yet been attained, Nagios continues to use the regular check interval. The value is specified in minutes. |
| Maximum number of check attempts | If the check returns a not *OK* status, the number of tests specified here is waited before the contact persons are notified. If the service reattains |

| Attribute | Description |
|---|---|
| | the *OK* status again before reaching the limit specified here, the internal counter is reset and there is no notification.<br><br>**Note**<br><br>The time delay for a notification is arranged both according to the *maximum number of check attempts* and to the *retry check interval*. At a *retry check interval* of two minutes and a *maximum number of check attempts* of 10, the first notification is performed after 20 minutes. |
| Check period | It is possible to specify a test period in order to impose time restrictions on a service test. There are no tests outside this period of time and consequently also no notifications. This can be useful for devices or services, which are deactivated at night. |

**Table 15.3. 'Notification' tab (advanced settings)**

| Attribute | Description |
|---|---|
| Notification interval | If an error occurs for a service, the contact persons are repeatedly notified in the interval specified here. A value of 0 deactivates the repeated notification. The value is specified in minutes. For example, if an interval of 240 were set, a notification would be sent every four hours. |
| Notification period | Notifications are only sent to the contact persons during the period specified here. If a service changes to the not-*OK* status outside of the period specified, the first notification is only sent once the specified period is reached, assuming the not-*OK* status continues that long.<br><br>**Note**<br><br>Notifications of errors which begin and end outside of the specified period are not repeated. |
| Notify if service state changes to WARNING | Configures whether a notification is sent when the service status changes to *WARNING* (see Section 15.2.1). |
| Notify if service state changes to CRITICAL | Configures whether a notification is sent when the service status changes to *CRITICAL* (see Section 15.2.1). |
| Notify if service state changes to UNREACHABLE | If a computer object is subordinate to another object (see Section 15.2.3.3), the status can no longer be requested in the case of error. This option can be used to configure whether a notification is triggered. |
| Notify if service state changes to RECOVERED | Configures whether a notification is sent when an error/warning/unaccessibility status is corrected to normal status. Notifications are only sent when the "RECOVERED" status is attained if a notification was sent for the original problem ("WARNING"/"CRITICAL"/"UNREACHABLE") in advance. |

**Table 15.4. 'Hosts' tab**

| Attribute | Description |
|---|---|
| Assigned hosts | The service test is performed for/on the computers assigned here. |

### 15.2.3.2. Configuration of a monitoring time period

Feedback

Nagios period objects are used by Nagios services to specify periods in which the service test should be performed or contact persons should be notified. Specification of the period is performed separately for each weekday.

Nagios services are administrated in the UMC module *Nagios* with the **Nagios time period** object type (see Section 4.4).

Nagios has no LDAP interface for the monitoring configuration, instead the configuration files are generated by a listener module when adding/removing/editing a Nagios time period.

Three standard periods are set up during the installation. The automatically created periods can be altered or deleted manually. However, they are used by the automatically created Nagios services to some extent. It is thus important to note that it is only possible to delete a Nagios period once it is no longer employed by any Nagios services:

| **Nagios time period** | **Description** |
|---|---|
| *24x7* | This object defines a period starting on Monday at 0:00 and ending on Sunday at 24:00 without any interruptions. |
| *WorkHours* | Defines the period from 8 a.m. to 4 p.m. from Monday to Friday respectively. |
| *NonWorkHours* | The opposite to the Nagios period *WorkHours*, this period covers the time from midnight to 8 a.m. and from 4 p.m. to midnight from Monday to Friday respectively and from 0:00 to 24:00 on Saturday and Sunday. |

**Table 15.5. 'General' tab**

| **Attribute** | **Description** |
|---|---|
| Name | An unambiguous name for the Nagios time period. |
| Description | Any description. |
| Monday - Sunday | This field contains a list of time periods. If there should be no period defined for a weekday, this weekday field should be left empty. The entry of the period always requires two-figure hour and minute entries separated by a colon. Start and end points are separated by a hyphen. If several periods are to be defined for one weekday, these can be entered in the text field separated by a comma. A whole day is represented by the period *00:00-24:00*, e.g., *08:00-12:00,12:45-17:00*. |

### 15.2.3.3. Assignment of Nagios checks to computers

Feedback

All the computer objects that can be administrated with Univention Management Console can be monitored with Nagios. Nagios services can only be assigned to a computer object if an IP address and a corresponding entry for the DNS forward zone are specified for it. The **Nagios** option must be switched on on the computer object in question to be able to activate the Nagios support. After activation there are two additional groups of input fields available beneath the tab **Advanced settings**. These can be used to assign the Nagios services conveniently among other things.

**Figure 15.5. Assigning Nagios checks to a host**

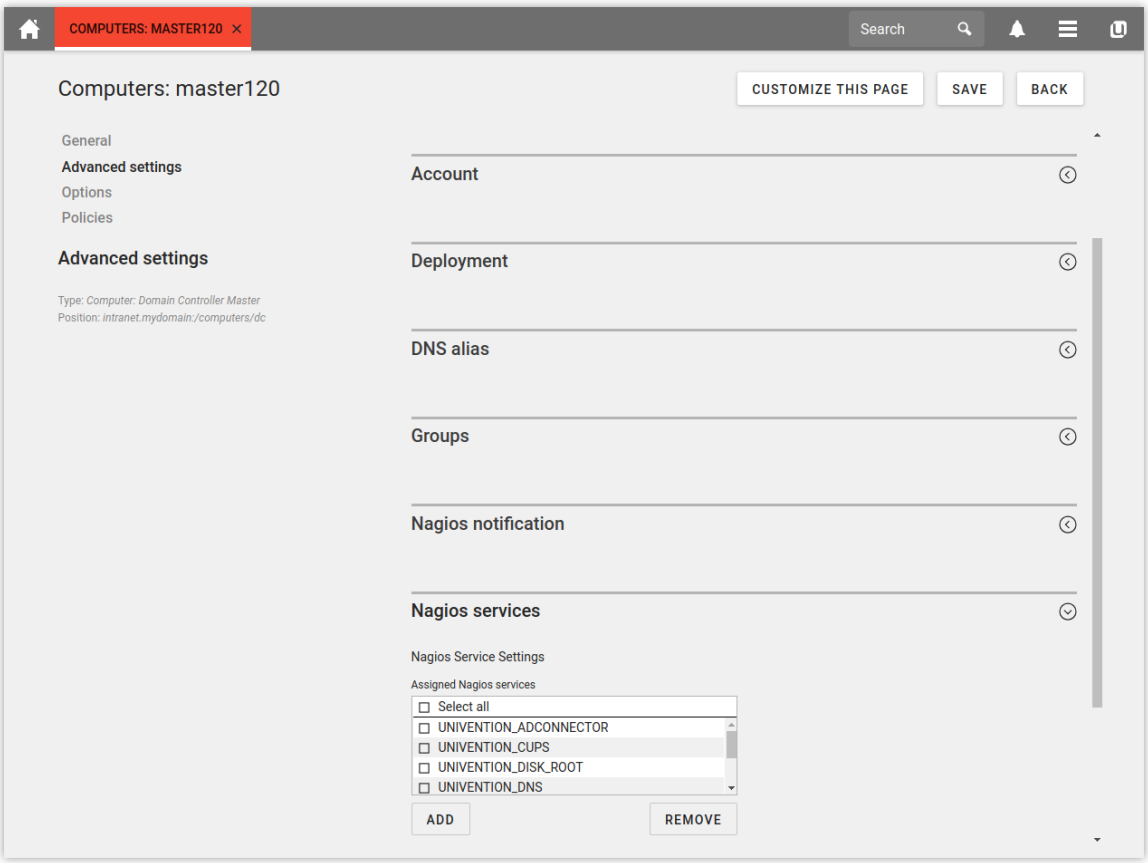



**Table 15.6. 'Nagios services' tab (advanced settings)**

| Attribute | Description |
|---|---|
| Assigned Nagios services | All the Nagios services that are checked for the current computer are listed here.<br><br>Parallel to this, the assignment of computers on the Nagios service object is also possible. |

**Table 15.7. 'Nagios notification' tab (advanced settings)**

| Attribute | Description |
|---|---|
| Email addresses of Nagios contacts | This list contains the e-mail address of contact persons who should be notified in the case of a problem. If no e-mail addresses are specified here, the local `root` user is notified. |
| Parent hosts | The entry of superordinate computers can be used to define dependencies between computers. Nagios continually tests whether the individual computers can be accessed. Should a superordinate computer not be accessible, no notifications of service faults are sent to the subordinate computer. Nagios also uses the specified dependencies in the user interface for graphic display. |

| Attribute | Description |
|---|---|
| | **Note**<br><br>No loops must occur when the superordinate computers are entered. In that case, the Nagios server would not adopt the new configuration and not be able to be started. |

### 15.2.3.4. Integration of additional Nagios plugin configurations

Feedback

If you wish to add expansions to the Nagios server configuration files created by the listener module, the manually created configuration files can be stored in the `/etc/nagios/conf.local.d/` directory. The added configuration files are only taken into account after the next restart of the server.

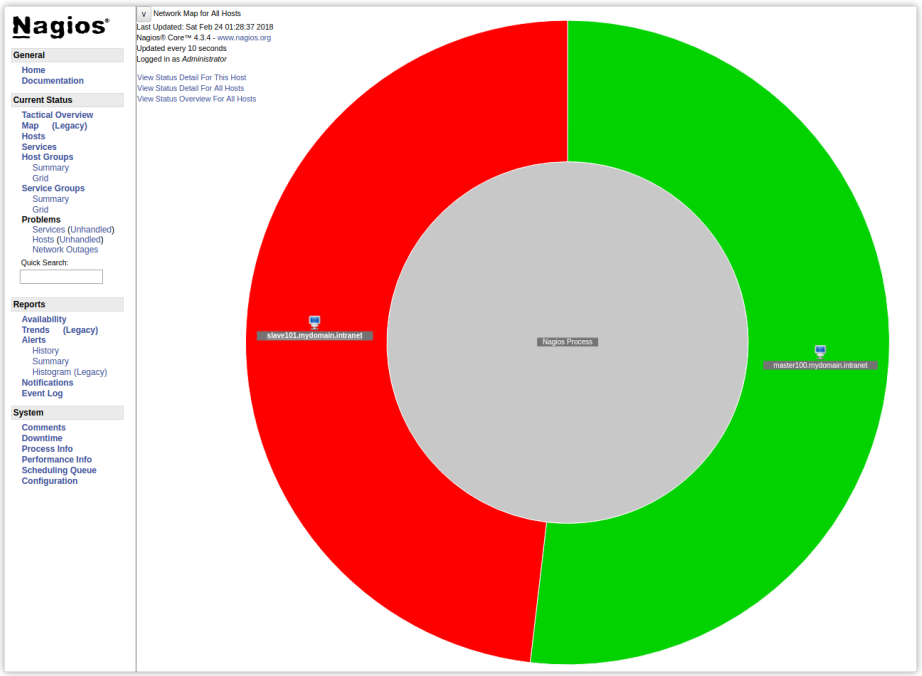
Expansions to the NRPE configurations can be stored in the `/etc/nagios/nrpe.local.d/` directory. Changes are only applied after the next restart of the Nagios NRPE Daemon.

## 15.2.4. Querying the system status via the Nagios web interface

Feedback

The Nagios interface is linked on the system overview page (see Section 4.2) under **Nagios** and can be opened directly at `http://SERVERNAME-OR-IP/nagios/`.

**Figure 15.6. Nagios status overview**




Access is only granted for users in the `Domain Admins` group (e.g., the Administrator) in the default setting. There is also the possibility of expanding the circle of those authorized to log in.

## 15.2.5. Integration of additional plugins

Feedback

The preconfigured Nagios plugins supplied with UCS can be complemented with additional plugins. A variety of available modules can be found at https://exchange.nagios.org/.

This section describes the integration of an external plugin taking the plugin `check_e2fs_next_fsck` as an example. The plugin checks whether a file system check is scheduled and emits a warning if one is scheduled within seven days and an error status if a file system check is scheduled for the next reboot.

The installation differs depending on whether the plugin is run via NRPE or not:

- If the plugin is run via NRPE, it must be copied into the `/usr/lib/nagios/plugins/` directory on all Nagios servers and on all the systems to be checked.
- If the plugin does not require local access, it need only be copied into the `/usr/lib/nagios/plugins/` directory on the Nagios server(s).

The plugin must be marked as an executable file (`chmod a+x PLUGIN`).

Some plugins are exclusively written in Perl, Python or Shell and do not require any external libraries or programs. These interpreters are always installed on all UCS systems. In contrast, if the plugin uses external programs or libraries, it must be ensured that these are installed on all the systems to be checked (for NRPE plugins) or on the Nagios servers (for remote plugins).

The Nagios plugin must now be registered. This is done using a macro in the `/etc/nagios-plugins/config/` directory. For this, a file such as `local.cfg` can be used, in which all the locally registered plugins are entered. The following example registers the plugin `check_e2fs_next_fsck`:

```
define command{
   command_name   check_fsck
   command_line   /usr/lib/nagios/plugins/check_e2fs_next_fsck
 }
```

Many plugins also use parameters to configure the thresholds for warnings and errors. These are determined in the *command_line* line. Similarly to the plugin itself, the macro file must also be copied onto all the systems to be monitored when using NRPE. The plugins, macros and any dependencies can also be bundled in a Debian package. Further information is available in [developer-reference].

The Nagios service must now be restarted:

```
/etc/init.d/nagios restart
```

The new plugin then only needs to be registered in Univention Management Console as a **Nagios service**, see Section 15.2.3.1. The name registered under *command_name* in the macro file must be entered as the **Plugin command**, in this example `check_fsck`, and the option **Use NRPE** enabled. The newly registered service can now be assigned to individual systems, see Section 15.2.3.3.

# Chapter 16. Virtualization

16.1. Introduction ..... 285
16.2. Installation ..... 285
16.3. Creating connections to cloud computing instances ..... 286
16.3.1. Creating an OpenStack connection ..... 287
16.3.2. Creating an EC2 connection ..... 289
16.4. Managing virtual machines with Univention Management Console ..... 289
16.4.1. Operations (Starting/stopping/suspending/deleting/migrating/cloning virtual machines) ... 290
16.4.2. Creating a virtual machine via a cloud connection ..... 292
16.4.3. Editing a virtual machine via a cloud connection ..... 292
16.4.4. Creating a virtual instance ..... 292
16.4.5. Modifying virtual machines ..... 293
16.5. KVM related UVMM features ..... 295
16.5.1. Image files of virtual machines ..... 295
16.5.2. Storage pools ..... 296
16.5.2.1. Accessing the default storage pool through a file share ..... 296
16.5.2.2. Adding a storage pool ..... 297
16.5.2.3. Moving the default storage pool ..... 297
16.5.3. CD/DVD/floppy drives in virtual machines ..... 298
16.5.4. Network interfaces in virtual instances ..... 298
16.5.5. Paravirtualization (virtIO) drivers for Microsoft Windows systems ..... 299
16.5.5.1. Installation of the virtIO drivers for KVM instances ..... 299
16.5.6. Snapshots ..... 299
16.5.7. Migration of virtual instances ..... 299
16.5.7.1. Migration of virtual machines from failed virtualization servers ..... 300
16.5.7.2. Migration of virtual machines between hosts with different CPUs ..... 300
16.6. Profiles ..... 301
16.6.1. Changing default network ..... 301

## 16.1. Introduction

Feedback

UCS Virtual Machine Manager (UVMM) is a tool for the administration of hybrid cloud environments. It allows central monitoring and administration of KVM virtualization servers registered in the UCS domain and virtual machines operated on it. In addition, virtual machines can be administered in OpenStack or EC2 environments. The administration is performed via the Univention Management Console module *Virtual machines*.

In principle, any operating system can be used on the virtualized systems.

## 16.2. Installation

Feedback

UCS Virtual Machine Manager can be installed from the Univention App Center with the application *UCS Virtual Machine Manager*. Alternatively, the software package ***univention-virtual-machine-manager-daemon*** can be installed. Additional information can be found in Section 5.6.

Administration of OpenStack cloud instances is possible directly after installation of the application with the Univention Management Console module *Virtual machines (UVMM)*. The *Amazon EC2 Cloud Connection* application needs to be installed for the administration of virtual machines in the Amazon EC2 cloud.

To add a KVM virtualization server for the administration via UCS Virtual Machine Manager locally, the *KVM virtualization server* application must be installed on a server of the domain from the Univention App

Center. The application can also be selected directly during the installation of a new UCS server. Alternatively, the software package ***univention-virtual-machine-manager-node-kvm*** can be installed.

CPU virtualization support is mandatory for the operation of KVM. This is provided by almost all current x86 CPUs. For more information, consult the KVM project website: http://www.linux-kvm.org/.

Additionally, the architecture must also be taken into account during installation of a virtualization server. 64-bit systems can only be virtualized on UCS systems which are installed using the amd64 architecture. A 64-bit system (amd64) is recommended for use as the virtualization server.

## 16.3. Creating connections to cloud computing instances

Feedback

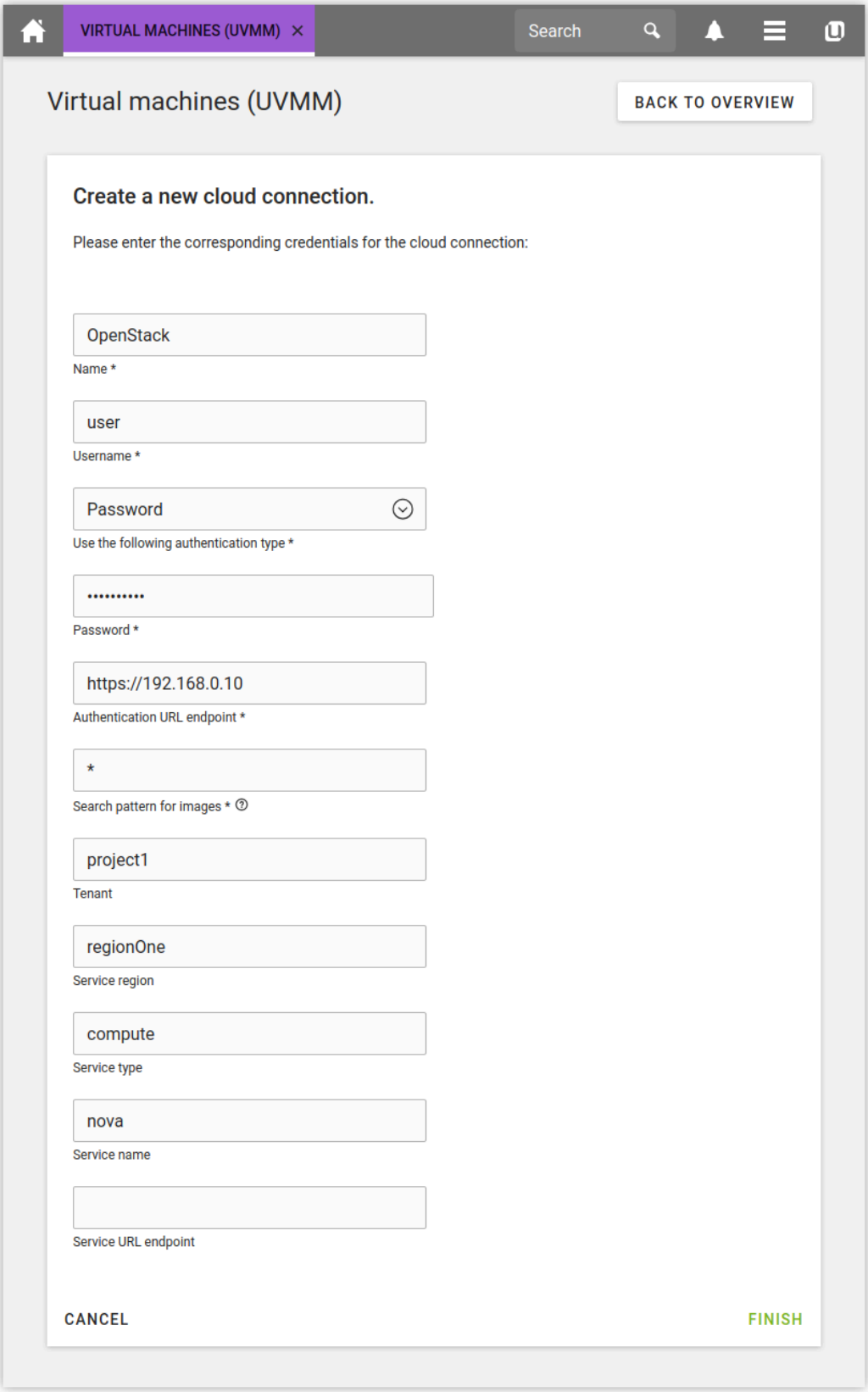
UCS Virtual Machine Manager supports connections to OpenStack. Installation of the application *Amazon EC2 cloud connection* makes administration of virtual machines on the Amazon EC2 cloud possible.

To create a new connection, the Univention Management Console module *Virtual machines (UVMM)* must be opened. Clicking on **Create** opens a wizard in which the **Create a new cloud connection** entry needs to be selected. In the drop-down field that appears you can now select the type of connection. Clicking on **Next** starts the set-up wizard. Once you have made the settings, clicking on **Finish** creates the connection. If an error occurs, it is displayed and the connection settings can be corrected. If the connection is established successfully, a wait animation is shown while the connection-specific information for the cloud connection is loaded. This covers for example the existing instances and available images for creating new instances.

## 16.3.1. Creating an OpenStack connection



**Figure 16.1. Creating a new connection to an OpenStack instance**



The following settings need to be made in the set-up wizard for creating a connection to an OpenStack instance:

**Table 16.1. Fields when setting up an OpenStack connection**

| Attribute | Description |
|---|---|
| Name | Sets the name of the connection. This will later be shown in the tree view of the Univention Management Console module. |
| Username | The user name to be used for authentication for OpenStack. |
| Use the following authentication type | There are two options to choose from. The corresponding value is entered in the field below.<br><br>Password<br><br>The password corresponding to the user name.<br><br>API key<br><br>The API key that allows the user access. |
| Authentication URL endpoint | The URL under which the authentication end point of the OpenStack instance can be reached should be entered here. If you want to establish an encrypted connection, the URL should be entered in the form `https://[...]`. As the public certificate for the OpenStack instance is used for the encrypted connection, this certificate needs to be made available on the UCS system on which the *UCS Virtual Machine Manager* application is installed. To this end, the public certificate must be copied into the `/usr/local/share/ca-certificates/` directory on the UCS server in PEM encryption and furnished with the suffix `.crt`. The following commands convert a certificate into the correct encryption and make the certificate known.<br><br>`openssl x509 -in [path/to/openstack-certificate] \`<br>`-outform pem -out /usr/local/share/ca-certificates/openstack.crt`<br><br>`update-ca-certificates`<br><br>The public certificate of the OpenStack authentication end point should be taken from the configuration of the OpenStack instance. The corresponding value to the certificate's path can be found under *ca_certs* in `keystone.conf`. |
| Search pattern for images | To create a new virtual machine, only the images that correspond to the configured search template are used as source images. The default value "*" (asterisk) is used to show all available images. |
| Project / tenant | The project or tenant name assigned to the user within the OpenStack environment. |
| Service region | The name of the region in which the user should work. The OpenStack default value is **`regionOne`**. |
| Service type | The type of the service under which the cloud compute function is available. The default value is **`compute`**. |
| Service | The name of the service under which the cloud compute function is available. The default value is **`nova`**. |

| Attribute | Description |
|---|---|
| Service URL endpoint | Optional value: The URL of the service end point is normally determined automatically when the user logs on to OpenStack. Should automatically determination not be possible, the corresponding URL can be entered here. |

## 16.3.2. Creating an EC2 connection

Feedback

The following settings need to be made in the set-up wizard for creating a connection to Amazon EC2:

**Table 16.2. Fields when setting up an Amazon EC2 connection**

| Attribute | Description |
|---|---|
| Name | Sets the name of the connection This will later be shown in the tree view of the Univention Management Console module. |
| EC2 region | Here you select the EC2 region to which you want to create the connection. Virtual machines are always assigned to precisely one region and not visible in other regions. The selection of available images can also vary depending on the region. Univention UCS images are available in all supported regions. |
| Access Key ID | The access key ID assigned to the Amazon EC2 account is comparable with a user name. |
| Secret Access Key | The secret access key for access via the Amazon EC2 account is comparable with a password. |
| Search pattern for AMIs | Image files used as a source for new instances are referred to as AMIs. The search filter specified here restricts the display of selectable AMIs when creating a new virtual instance. The default value "*" (asterisk) is used to show all available images. |

# 16.4. Managing virtual machines with Univention Management Console

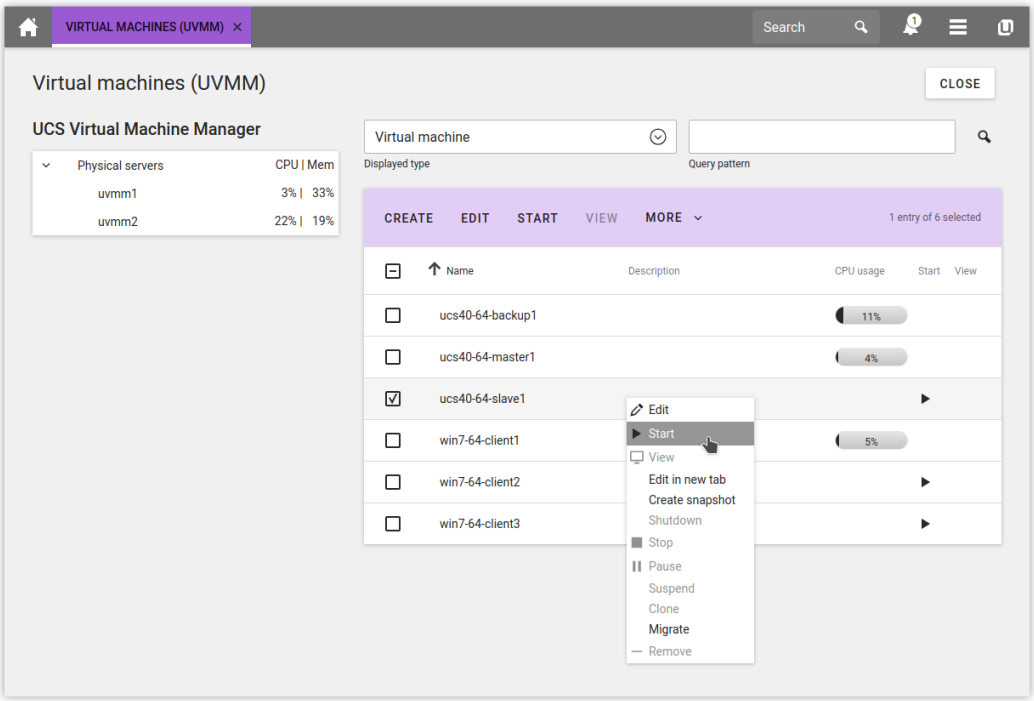
Feedback

The UMC module *Virtual machines (UVMM)* offers the possibility to create, edit and delete virtual instances/machines and to change their status. In principle, these functions are independent of the virtualization technology employed (KVM or cloud-based), however they may vary slightly depending on the hypervisor in use. The items that must be observed are illustrated in the following section on the description of the functions.

# 16.4.1. Operations (Starting/stopping/suspending/deleting/migrating/cloning virtual machines)

Feedback

**Figure 16.2. Overview of virtual machines**




In the main dialogue of the UMC module, a tree structure is displayed on the left-hand side, which gives an overview of the existing virtualization servers. All the virtual machines are listed in the right half of the screen. If one clicks on the name of a virtualization server, only the instances of that server are listed. The search mask can also be used to search for individual virtual machines.

In the overview of the virtual machines, the computer icon shows the state a virtual machine is in, e.g., whether it is running (computer symbol with green arrow), paused (computer symbol with yellow line) or stopped (computer without additional symbol). Virtual machines in cloud computing environments can also be depicted as deleted (computer with red cross) or as pending (computer with hourglass).

The icon showing an arrow pointing right can be used to start a virtual instance.

Running instances can be accessed via the VNC protocol - insofar as this is configured. The icon with the stylized screen opens a connection with noVNC, a HTML5-based client. Any other VNC client can also be used for the access; the VNC port is displayed in a tooltip above the computer name.

The **more** choice box can be used to perform other operations: The following operations are available on running instances:

Stop

: turns the virtual machine off. It must be noted that the operating system of the virtual machine is not shutdown first, i.e., it should be compared with turning off a computer by pulling the power plug.

Pause

: assigns the instance no further CPU time. This still uses the working memory on the virtualization server, but the instance itself is paused.

Suspend

saves the contents of the machine's system memory on the hard drive and does not assign the machine further CPU time, i.e., compared with **Pause** the working memory is also freed. This function is only available on KVM-based virtualization servers.

Migrate

migrates the virtual machine to another virtualization server. Further information can be found in Section 16.5.7.

The following operations are available on saved or stopped instances:

Remove

Virtual instances no longer required can be deleted along with all their hard drives and ISO images. The images to be deleted can be selected from a list. It must be noted that ISO images and sometimes also hard drive images may still be used by other instances. They should only be deleted when they are no longer used by any instance.

Migrate

migrates the virtual machine to another virtualization server. Further information can be found in Section 16.5.7.

Clone

creates a copy of the current VM. It is given a freely selectable, new name. Network interfaces are adopted, but can also alternatively be randomly regenerated. Mounted CD and DVD drives from the source VM are also integrated in the clone, while hard drives are copied insofar as the storage pool supports the copying. Snapshots are not copied!

The following operations are available for virtual machines operated in cloud-based environments.

Restart (hard)

Restarts the virtual machine as if the reset button had been pressed. This can result in data loss.

Restart (soft)

Sends an ACPI reset event to the virtual machine. If the operating system of the virtual machine interprets this correctly, a regular restart is performed.

Shutdown (soft)

Sends an ACPI shutdown event to the virtual machine. If the operating system of the virtual machine interprets this correctly, it is shut down and turned off regularly.

Pause

The machine is not assigned any more CPU time. This still uses the working memory on the physical host, but the machine itself is paused.

Suspend

Saves the contents of the machine's working memory on the hard drive memory and does not assign the machine further CPU time, i.e., compared with **Pause**, the working memory is also freed.

Delete

Turns the virtual machine off and deletes all the corresponding data permanently.

## 16.4.2. Creating a virtual machine via a cloud connection

Feedback

Virtual machines in cloud-based virtualization environments can be created in just a few steps in UVMM using the wizard by clicking on **Create**.

In the **Create a virtual machine or a cloud connection** input mask you can select the cloud connection via which you wish to create the virtual machine. Once a connection has been selected and you have clicked on **Next**, the wizard for creating a new virtual machine opens. Once the parameters have been set, the new virtual machine is created by clicking on **Finish**.

**Table 16.3. Creating a virtual machine via a cloud connection**

| Attribute | Description |
|---|---|
| Name | Defines the name of the virtual machine |
| Choose a source image / source AMI | The initial status of a virtual machine when created is specified via a source image (OpenStack) or source AMI (EC2). This type of image usually includes a prepared operating systems that the user can customize after the start-up. Any number of virtual machines can be created from one source image. |
| Choose an instance size | An instance size is assigned to a virtual machine when it is created. This is composed of available memory and the size of the available hard drive memory. When a virtual machine is created in an OpenStack environment, the number of the CPU cores is also determined when selecting the size. |
| Select a key pair | To allow safe access to the virtual machine via ssh, an ssh key for configuration of the root account is added to the machine the first time it is started. With this key, it is possible to access the machine via ssh without a password. For this to happen, there must be access to the private key part of the key pair. The access to the instance can be performed with the following command, for example, if the instance is running:<br>`ssh -i [path/to/private/key] root@[instance-ip-address]` |
| Configure security group | This setting configures which security group is set for the new virtual machine. A security group determines which TCP ports are approved for external access to a virtual machine. |

## 16.4.3. Editing a virtual machine via a cloud connection

Feedback

By selecting a virtual machine and clicking on **Edit** you can view the configured settings of the virtual machine on a separate page. The IP address via which the virtual machine can be reached is shown here in particular.

## 16.4.4. Creating a virtual instance

Feedback

Virtual machines on local KVM servers can be created with the assistant in a few steps in UVMM by clicking on **Create**.

In the **Create a virtual machine or a cloud connection** input mask you can select the virtualization server on which you wish to create the virtual machine. If a KVM virtualization server is selected here and **Continue**

clicked, the machine profile selection page opens. The selection of the **Profile** specifies some of the basic settings for the virtual instance (see Section 16.6).

The virtual machine is now given a **Name** and an optional **Description** and assigned **Memory** and **CPUs**. The **Enable direct access** option specifies whether the machine can be accessed via the VNC protocol. This is generally required for the initial operating system installation.

Now the disk drives of the virtual machines are configured. The setup is documented in Section 16.5.1.

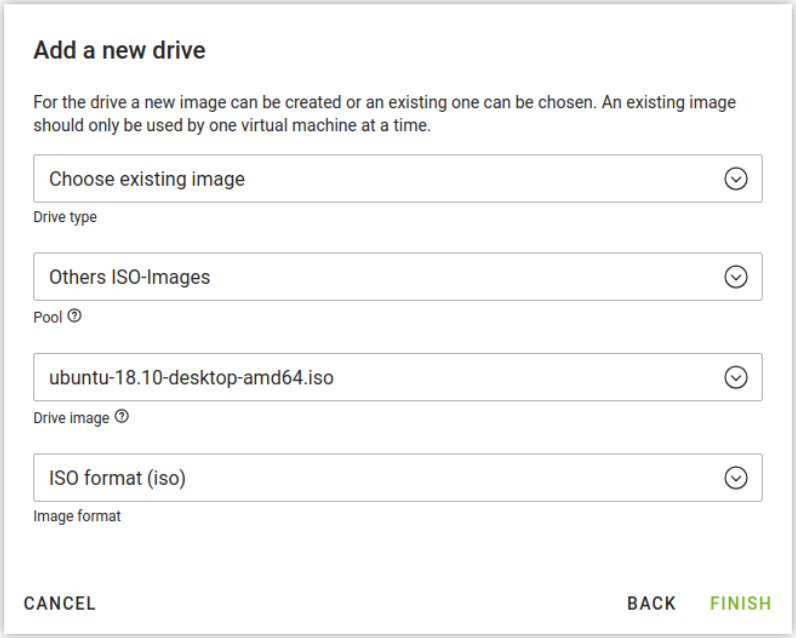
Clicking **Finish** concludes the creation of the virtual machine.

## 16.4.5. Modifying virtual machines

Feedback

In the overview list, a virtual machine can be edited by clicking on the icon with the stylized pen.

### Figure 16.3. Modifying the settings of a DVD drive




Most settings of a virtual machine can only be changed if it is turned off.

### Table 16.4. 'General' tab

| Attribute | Description |
|---|---|
| Name | Defines the name of the virtual machine. This does not have to be the same as the name of the host in the LDAP directory. |
| Operating system | The operating system installed in the virtual instance. Any text can be entered here. |
| Contact | Defines the contact person for the virtual machine. If an e-mail address is specified here, an external e-mail program can then be run via the mouseover that appears. |
| Description | Can be used to describe the function of the virtual machine, e.g. *mail server* or it's state. The description is shown in the overview of the virtual machines as a mouseover. |

The tab **Devices** allows the configuration of drives and network interfaces. An introduction to the supported devices, image formats and storage pools can be found in the Section 16.5.1. An introduction to the supported network card settings can be found in the Section 16.5.4.

**Drives** lists all existing drives, the image files used, their size and the assigned storage pools. One can click on the stylized minus sign to delete a drive and **Edit** can be used to adjust setting subsequently.

**Paravirtual drive** allows specification of whether the access to the drive should be paravirtualized. Where possible, this setting should not be changed for a virtual machine which already has an operating system installed, as this may disrupt the access of partitions.

If drives or network interfaces are subsequently added to a virtual instance, the utilization of paravirtualization is determined by heuristics or its profile.

**Add drive** can be used to add an additional drive.

This menu contains a list of all network cards; in addition, new cards can be added or existing ones edited. **Add network interface** can be used to add another virtual network card.

The tab **Snapshots** contains a list of all available snapshots. An introduction to snapshots can be found in the Section 16.5.6.

**Snapshots** includes a list of all the existing snapshots. **Resume** can be used to restore an earlier status.

### Caution

The current machine state is lost if the old snapshot is restored. However, there is no reason not to save the current state in an additional snapshot in advance.

A snapshot can be removed by clicking in the stylized minus sign. The current state of the virtual machine is not modified by this.

**Create new snapshot** can be used to create a snapshot with the name of your choice, e.g., *DC Master before update to UCS 4.0-1*. In addition to the description the time is saved when the snapshot is created.

The tab **Migration targethosts** can be used to configure the host systems, to which the virtual machine can be migrated to. Further information on migration can be found in Section 16.5.7.

**Table 16.5. 'Advanced' tab**

| Attribute | Description |
|---|---|
| Architecture | Specifies the architecture of the emulated hardware. It must be noted that virtual 64-bit machines can only be created on virtualization servers using the amd64 architecture. This setting is not shown on i386 systems. |
| Number of CPUs | Defines how many CPU sockets are assigned to the virtual instance. The number of NUMA nodes, cores and CPU threads is not currently configurable. |
| CPU model | The CPU model for the virtual machine. The list of usable models depends on the concrete host system. A complete list of available models is available on the command line running `virsh domcapabilities`. Further information - especially on live migration - see Section 16.5.7. |
| Memory | Specifies the size of the assigned system memory. |
| Virtualization technology | The technology used for virtualization. This setting can only be specified when creating a virtual instance. |

| Attribute | Description |
|---|---|
| RTC reference | In fully virtualized systems, a computer clock is emulated for each virtual machine (paravirtualized systems access the clock on the host system directly). This option controls the format of the emulated clock; it an either be saved in the **coordinated universal time (UTC)** or the **local timezone**. The use of UTC is recommended for Linux system and the use of the local time zone recommended for Microsoft Windows systems. |
| Boot order | Specifies the order in which the emulated BIOS of the virtual machine searches the drives for bootable media. This setting is only available for fully-virtualized instances. On paravirtualized machines it is only possible to select one hard drive from which the kernel should be used. |
| Enable Hyper-V Enlightenment | Allow guest systems like `Microsoft Windows` to run more efficient as virtual machines. |
| Always start VM with host | Defines whether the virtual machine is automatically started when the host systems itself starts up. |
| Direct access (VNC) | Defines whether VNC access to the virtual machine is available. If the option is enabled, the virtual machine can be accessed directly via the UMC module using an HTML5-based VNC client or any other VNC client. The VNC URL is displayed in a tool tip. |
| Globally available | This allows VNC access from other systems than the virtualization server. |
| VNC Password | Sets a password for the VNC connection. |
| Keyboard layout | Defines the layout for the keyboard in the VNC session. |

# 16.5. KVM related UVMM features

Feedback

## 16.5.1. Image files of virtual machines

Feedback

If virtual hard drives are added to an instance, *image files* are usually used for the data keeping. An image file can either be generated for this purpose or an existing image file can be assigned to a virtual machine. Alternatively, a native block device (hard drive partition, logical volume, iSCSI volume) can be assigned to a virtual machine. The direct use of block devices offers performance advantages and is less susceptible to computer crashes.

Hard drive images can be administrated in two ways on KVM systems; by default images are saved in the **Extended format (qcow2)**. This format supports Copy-on-write which means that changes do not overwrite the original version, but store new versions in different locations. The internal references of the file administration are then updated to allow both access to the original and the new version. Snapshots can only be created when using hard drive images in **Extended format**. Alternatively, you can also access a hard drive image in **Simple format (raw)**.

Operating systems use a so-called *page cache* to accelerate accesses to storage media. If data are accessed which have already been read off a hard drive and these data are still present in the cache, the comparatively slow access to the storage medium is not necessary and the request is answered directly from the page cache.

Write accesses are generally also not directly written on the hard drive, but are usually bundled and, consequently, written more efficiently. However, this involves the risk of data loss, if, for example, a system crashes or the power supply is interrupted. The data which have been only saved in the write cache up to that point and have yet to be synchronized on the storage medium are lost. For this reasons, modern operating systems generally only keep pending write changes for a maximum of several seconds before writing them to the hard drive.

In order to avoid data being stored doubly in the page cache of the host system and also of the guest system, cache strategies can be configured with the **Caching** option when using KVM, which influence the use of the host system's page cache:

- The default setting since UCS-3.1 is **none**: in this setting, KVM accesses the hard drive directly and bypasses the page cache on the virtualization server. Read accesses are answered directly by the hard drive every time and write accesses are passed directly on to the hard drive.
- The **write-through** strategy uses the page cache on the virtualization server, but every write access is also passed on directly to the storage medium. On virtualization servers with a lot of free system memory, read accesses can be more efficient than **none**. However, the double caching generally has a negative effect on the overall performance. [1]
- If the **write-back** strategy is used, the host's page cache will be used for both read and write accesses. Write accesses are initially only performed in the page cache, before they are then written to the hard drive at a later point in time. In this case, if the host system crashes, data may be lost.
- With the **unsafe** strategy, synchronization requests sent by the guest system are ignored in order to force the writing of outstanding data on the storage medium explicitly. Compared with **write-back**, this once again increases the performance, but can result in data loss if the host system crashes. This version is only practical for test systems or comparable installations in which data loss due to the crashing of the host system is not dramatic.
- The **directsync** strategy corresponds to **none**, with the only difference being that here synchronization is explicitly forced after every write access.
- The **Hypervisor default** option is dependent on the UCS version and the KVM version with which a guest system was installed: Originally, the standard value until UCS 3.0 was implicitly **write-through**, but KVM was modified to such an extent with UCS 3.1 that **none** is now used for all old VMs instead. For VMs re-saved with UCS 3.1 the standard value is implicitly **write-through** again, but new VMs are explicitly saved with **none**.

If a live migration of virtual machines between different virtualization servers is planned, the storage pool must be stored on a system which can be accessed by all virtualization servers (e.g., an NFS share or an iSCSI target). This is described in Section 16.5.2.

Image files are created as sparse files with the specified size, i.e., these files only grow when they are used and then up to the maximally specified size and thus initially require only minimal disk space. As there is a risk here of the disk space being used up during operation, a Nagios monitoring should be integrated, see Section 15.2.

Where possible, hard drive images should be configured paravirtualized. In the case of UCS systems installed virtualized in KVM, a paravirtualized access is activated automatically when the UCS profile is selected. The configuration of Microsoft Windows systems is documented in Section 16.5.5.

## 16.5.2. Storage pools

Feedback

These image files are stored in so-called storage pools. They can either be stored locally on the virtualization server or on a file share. The connection of a storage pool over iSCSI is documented in [ext-doc-uvmm].

### 16.5.2.1. Accessing the default storage pool through a file share

Feedback

Each virtualization server provides a storage pool with the name *default* in the standard configuration. It can be found on the virtualization servers in the `/var/lib/libvirt/images/` directory.

[1] Instead, it is recommended to make the free memory directly available to the VMs so that they can use the additional memory more efficiently themselves, for instance for caching.

To allow simple access to the storage pool, you can set up a share for the `/var/lib/libvirt/images/` directory. To do so, you need to create a share with the following options in the UMC module **Shares**. The share can then be accessed easily from Windows clients via a CIFS network share (or via an NFS mount).

- General/General settings
  - Name: **`UVMM-Pool`**
  - Host: The hostname of the UVMM server
  - Directory: **`/var/lib/libvirt/images`**
  - Directory owner, Directory owner group and Directory mode can remain in the default setting
- Advanced settings/Samba permissions
  - Valid users or groups: **`Administrator`**

The image files of a virtual hard drive include all the user data of the virtualized system! The **Valid users or groups** option ensures that, irrespective of the file system permissions, only the Administrator user can access the share.

### 16.5.2.2. Adding a storage pool

Feedback

It is not possible to create an additional storage pool via Univention Management Console. Instead, this must be done by logging in to the virtualization server as the `root` user. The following steps are required for this:

- The directory in which the data from the storage pool are to be saved must be created; in this case `/mnt/storage/`.
- The following command is used to create the new *Testpool* storage pool:

```
virsh pool-define-as Testpool dir - - - - "/mnt/storage"
```

- The libvirt library used by UVMM differentiates between active and inactive storage pools. To be able to use the storage pool directly, it must be activated:

```
virsh pool-start Testpool
```

  The following command ensures that the pool is activated automatically the next time the system is started:

```
virsh pool-autostart Testpool
```

### 16.5.2.3. Moving the default storage pool

Feedback

To change the underlying file path of the default storage pool at a later point in time, one must log in to the virtualization server as the `root` user. The following steps are required for this:

- The Univention Configuration Registry variable `uvmm/pool/default/path` must be changed to the new directory.
- The following commands remove the old storage pool; the pool is changed over to the new path the next time the UVMM is restarted:

```
virsh pool-destroy default
virsh pool-undefine default
invoke-rc.d univention-virtual-machine-manager-daemon restart
```

```
invoke-rc.d univention-virtual-machine-manager-node-common restart
```

## 16.5.3. CD/DVD/floppy drives in virtual machines

Feedback

CD-/DVD-ROM/floppy drives can be mounted in two ways:

- An ISO image can be assigned from a storage pool. If no additional storage pool has been created, the files from the pool *default* are read from the directory `/var/lib/libvirt/images/`.
- Alternatively, a physical drive from the virtualization server can be connected with the virtual machine.

It is also possible to provide a virtual machine with a disk drive via an image (in `VFD` format) or the pass-through of a physical drive.

If drives are defined for a new virtual machine, it must be ensured that it is possible to boot from the CD-ROM drive. The UVMM profile (see Section 16.6) specifies the boot order for the fully-virtualized instances in advance. For the paravirtualized instances, it is defined by the order on the definition of the drives and can be adapted subsequently in the settings section.

## 16.5.4. Network interfaces in virtual instances

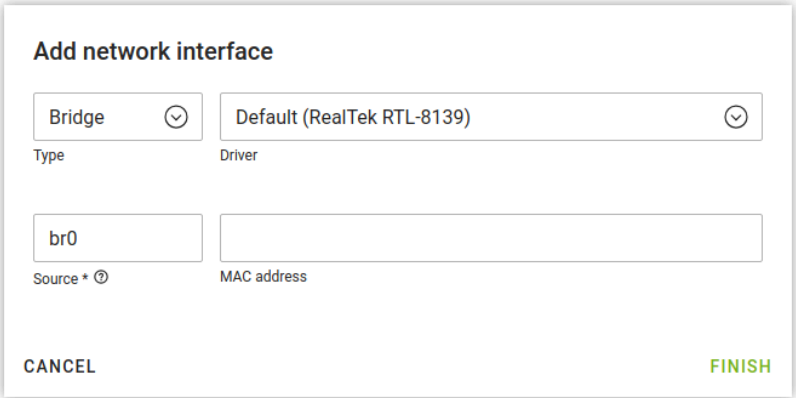
Feedback

When a virtual machine is created, it is automatically assigned a network card with a randomly generated MAC address. It can be subsequently changed.

Two types of network connections are possible:

- In the basic settings, a *Bridge* on the virtualization server is used to access the network directly. The virtual machine uses its own IP address and can thus also be reached from other computers.
- *Network Address Translation (NAT)* network cards are defined in a private network on the virtualization server. To do so, the virtual machine(s) must be assigned an IP address from the `192.0.2.0/24` network. This virtual instance is granted the access to the external network via NAT, so that the access is performed via the virtualization server's IP address. The virtual machine can thus not be reached from other computers, but can create all outgoing connections itself.

**Figure 16.4. Adding a virtual network interface**




The UVMM servers are already preconfigured for bridging and NAT. However, there are restrictions for bridged network cards which are described in Section 8.2.4.1.4.1. For each virtual machine the desired network can be selected through the **Source** setting.

NAT network cards are only restricted by the IP addresses available in the `192.0.2.0/24` network.

The **Driver** can be used to select what type of card will be provided. The **Realtek RTL-8139** is supported by almost all operating systems, the **Intel Pro-1000** offers advanced abilities and a **Paravirtual device** offers the best performance.

## 16.5.5. Paravirtualization (virtIO) drivers for Microsoft Windows systems

Feedback

KVM supports paravirtualization via the virtIO interface. The use of paravirtualization allows the virtualized systems direct access to the resources of the virtualization server. This considerably improves performance. We recommend the use of paravirtualization.

Current Linux systems support paravirtualization as standard. The installation of the KVM packages provides suitable images which can then be mounted in a virtual machine in the disk management. The images are integrated in the storage area specified by the Univention Configuration Registry variable `uvmm/pool/default/patht`. On KVM virtualization servers, there is an ISO image with the name *KVM Windows drivers*, which contains the virtIO virtualization drivers for Microsoft Windows.

### 16.5.5.1. Installation of the virtIO drivers for KVM instances

Feedback

In Windows systems installed under KVM, paravirtualization must be activated *before* beginning the Windows installation.

The virtIO interface allows the efficient usage of network and storage resources for a virtual machine on the KVM hypervisor. The following steps describe the installation of the virtIO drivers on Windows 7.

- A CD/DVD drive needs to be setup in the drive settings with the image **KVM Windows drivers** assigned.
- The hard disk drive has to be edited in the **Devices** menu in UVMM and the checkbox **Paravirtual drive** must be ticked.
- The **Driver** must be configured to **Paravirtual device (virtio)** for the network card(s).
- The initial steps during the installation of the Windows system take place as usual. A warning appears during hard disk partitioning and states that no mass storage could be found. This is not an error because the virtIO drivers are necessary for a paravirtualized device. The virtIO drivers can be installed in the same menu with **Load drivers**. The **Red Hat virtIO SCSI Controller** has to be chosen for Windows 7 (and for Windows 2003 and Windows 2008 respectively) and the **Red Hat virtIO Ethernet Adapter** for Windows 2008/Windows 7. After the device drivers have been installed, the mass storage is available in the Windows installer and the installation of Microsoft Windows can be continued.
- After completing the installation the devices `Red Hat virtIO SCSI Disk Device` and `Red Hat virtIO Ethernet Adapter` can be found in the Windows device manager.

## 16.5.6. Snapshots

Feedback

UVMM offers the possibility to save the contents of the main and hard drive memory of a virtual machine in snapshots. This allows the administrator to revert to these snapshots at a later point in time, which makes them a useful "safety net" when installing software updates.

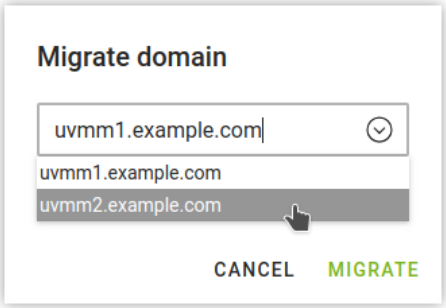
Snapshots can only be used with KVM instances which access all their hard drive images in Qcow2 format. All snapshots are stored using copy-on-write (see Section 16.4.4) directly in the hard drive image file.

## 16.5.7. Migration of virtual instances

Feedback

UVMM offers the the possibility of migrating a virtual machine to another virtualization server. This works with both paused and running instances (live migration). The option is only offered if at least two compatible virtualization servers are available in the domain.

**Figure 16.5. Migrating a virtual instance**




During the migration it must be noted that the images of the mounted hard drives and CD-ROM drive must be accessible by both virtualization servers. This can be achieved, for example, by storing the images in a central storage system. Notes on the setting up of this type of environment can be found under Section 16.5.2.

### 16.5.7.1. Migration of virtual machines from failed virtualization servers

Feedback

Information about the virtual machines running on the virtualization servers is stored centrally in the UCS Virtual Machine Manager. If a server fails (failure detection is performed periodically every 15 seconds), the server and the virtual instances operated on it are identified as inaccessible with a red symbol, a warning appears and **Migrate** is offered as the only operation in the menu.

Following the migration, the virtual instance is no longer displayed in the overview tree of the failed virtualization server in the UVMM.

#### Caution

It must be ensured under all circumstances that the virtual machine on the original and the secondary server are not started in parallel; this would involve their both writing in the image files simultaneously, which would result in data loss. If virtual machines are started automatically after startup, simultaneous access must be prohibited by disconnecting the network connection or restricting access to the storage pool.

If the failed computer is reactivated - e.g., in the case of a temporary power failure - the virtual machines remain available on the system locally and are reported to UVMM; consequently, there are then two versions of the instance.

As such, one of the two instances should subsequently be deleted. However, the employed image files for the drives should *not* be deleted at the same time.

### 16.5.7.2. Migration of virtual machines between hosts with different CPUs

Feedback

Virtual machines can be migrated between hosts with compatible CPUs. Newer CPUs are normally backwards compatible with previous generations of the CPU and only gain new features. The reverse is not true: If the guest operating system has decided to use a new feature and that feature does no longer work after migration, the virtual machine will crash.

By default, no specific CPU model is explicitly configured: Rather, the CPU functions of the respective virtualization server are passed directly to the virtual machine. The advantage is that the performance is higher, the disadvantage is that it can lead to crashes during live migration. To prevent migration between incompatible CPUs, UVMM can consider the CPU model of the server. This functionality has to be configured per virtual machine and only becomes effective after a restart of the virtual machine. A reboot of the running guest operating system is not sufficient; The virtual machine must be turned off if necessary and must be started again.

On the **Advanced** tab of the virtual machine the CPU model can be configured explicitly.

Using Univention Configuration Registry variable `uvmm/vm/cpu/host-model` the required customization of virtual machines can be automated. The following values can be used:

`missing`

UVMM activates the check for all virtual machines for which the CPU model is not already explicitly configured.

`always`

UVMM activates the check for all virtual machines, regardless of whether a CPU model is already explicitly configured or not. This will overwrite any previous CPU model configuration.

`remove`

UVMM removes any CPU model configurations.

- unset -

UVMM does not reconfigure any virtual machine. This is the default.

### Caution

If multiple UVMM daemons are used, Univention Configuration Registry variable `uvmm/vm/cpu/host-model` should be set identically on all UCS systems.

## 16.6. Profiles

Feedback

Profiles are used to store initial settings when creating new virtual machines. Among others this includes the following settings:

- name prefix for new virtual machines
- number of virtual CPUs
- CPU model
- default RAM size
- default size for new disk images
- default boot order for fully-virtualized virtual machines
- use of paravirtual device drivers
- default settings for direct access per VNC
- network bridge name

The existing UVMM profiles are stored in the LDAP directory and can also be edited there. The profiles can be found in the UMC module **LDAP directory** in the container `cn=Profiles,cn=Virtual Machine Manager`. Additional profiles can also be added there.

### 16.6.1. Changing default network

Feedback

The name of the bridge used as the default network interface is stored in UVMM profiles. If the default interface `br0` is changed, the name should be updated as well. The following command updates all profiles currently using interface *`$OLD`* to use the bridge *`$NEW`*:

```
udm uvmm/profile list --filter interface="$OLD" |
 sed -ne 's/^DN: //p' |
 xargs -r -d '\n' -n 1 udm uvmm/profile modify --set interface="$NEW" --
dn
```

# Chapter 17. Data backup with Bacula

17.1. Introduction ..... 303
17.2. Scope of backup on a UCS system ..... 304
17.3. Installation ..... 304
17.4. Configuration of the backup components ..... 305
17.4.1. Directory Daemon ..... 305
17.4.2. Storage ..... 305
17.4.3. File Daemon ..... 305
17.4.4. Bacula Console ..... 306
17.4.5. Firewall adjustments ..... 306
17.5. Configuration of the backup (interval, data, etc.) ..... 306
17.6. Administration via the Bacula console ..... 307
17.7. Backup of the catalog database ..... 308
17.8. Further information ..... 309

## 17.1. Introduction

Feedback

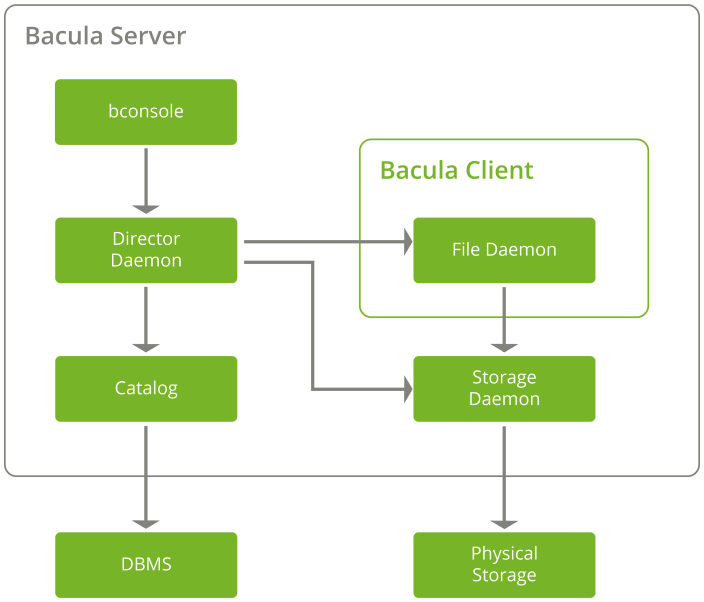
Bacula is a network-enabled data backup solution with a client/server architecture. It allows data backup and restore in heterogeneous environments. This chapter refers to the ***univention-bacula*** package which is delivered as a component of UCS. Other backup solutions can be selected and installed in the Univention App Center, including Bacula Enterprise.

Bacula is composed of a range of individual services and programs, which control the various aspects of the data backup:

- The *director daemon* is the central control unit in which most settings for backup and restore are saved. The remaining Bacula services are configured in the director.
- The *storage daemon* controls access to the backup media (e.g., a tape library or hard drive) and receives the instructions from the *director* about which systems should be backed up or restored.
- The *file daemon* is installed on the clients and receives the instructions of the *director* about which files should be backed up or restored via which *storage daemon*.
- The *catalog* saves all the backups in a database and allows the restore of individual files or directories.
- The *Bacula console* is the central user interface for the *director daemon*. The backup / restore jobs can be started here. It can also be used to perform administrative tasks - such as the integration of backup media - and requesting status information.
- The *Bacula administration tool* is a graphic version of the Bacula console.

The backup settings (data to be backed up, backup mode and times) are thus configured in the *director daemon* and the backup started automatically or via the *Bacula console*. The *file daemon* then supplies the data to be backed up to the *storage daemon*, which is responsible for saving the data on physical media. In addition, meta information concerning the backups are also saved in a database via the *catalog*.

**Figure 17.1. Bacula Schema**




# 17.2. Scope of backup on a UCS system

Feedback

If there is sufficient storage capacity available, it is recommended to back up the complete system. However, not all the data on a UCS system need to be backed up. For example, the program packages delivered with UCS are available after reinstallation anyway.

The following information gives an overview of a typical system. Deviations are possible depending on the software installed. This must be checked in each case and carefully tested with a test restore run!

The `/dev/`, `/proc/` and `/sys/` directories only contain files automatically generated by the kernel, so they do not need to be backed up.

These files should generally always be backed up: The `/home/` and `/root` directories contain user data, the configuration of the UCS system is backed up in `/etc/` and the `/var/` directory includes runtime data such as the mails of a mail server.

The `/bin/`, `/boot/`, `/lib/`, `/usr/` and `/sbin/` directories usually only include programs and data delivered with the UCS installation.

# 17.3. Installation

Feedback

In this documentation it is assumed that the *director daemon*, *storage daemon* and *catalog* are present on one a system, the Bacula server. These components are set up by installing the ***univention-bacula*** package.

The *file daemon* must be installed on all the systems on which data are to be backed up using the ***bacula-client*** package.

The storage of the catalog data is performed in a PostgreSQL database, which is created during installation. The access information for this database (database name, name/password of database user) are then available in the `/etc/dbconfig-common/bacula-director-pgsql.conf` file in the `dbc_dbpass` and `dbc_dbuser` fields.

# 17.4. Configuration of the backup components

Feedback

The configuration of the Bacula services is performed via various configuration files. The following text explains important options; further configuration options are described in the Bacula documentation.

## 17.4.1. Directory Daemon

Feedback

The directory daemon is managed via the *Director* section of the `/etc/bacula/bacula-dir.conf` file.

The default values can be kept, only the `DirAddress` option should be changed from `127.0.0.1`, in other words `localhost`, to the IP address of the Bacula server. In addition, the `Password` field should be configured

```
Director {
  Name = sec-dir
  DIRport = 9101
  QueryFile = "/etc/bacula/scripts/query.sql"
  WorkingDirectory = "/var/lib/bacula"
  PidDirectory = "/var/run/bacula"
  Maximum Concurrent Jobs = 1
  Password = "master-dir-password"
  Messages = Daemon
  DirAddress = 192.0.2.125
}
```

## 17.4.2. Storage

Feedback

The storage daemon is managed via the *Storage* section of the `/etc/bacula/bacula-sd.conf` file.

Here the default values can largely be retained; only the `SDAddress` option needs to be adapted to the IP address of the storage daemon.

```
Storage {
  Name = sec-sd
  SDPort = 9103
  WorkingDirectory = "/var/lib/bacula"
  Pid Directory = "/var/run/bacula"
  Maximum Concurrent Jobs = 20
  SDAddress = 192.0.2.125
}
```

The *Director* section refers to the Bacula server and a password is set that the server must use for access:

```
Director {
  Name = sec-dir
  Password = "master-storage-password"
}
```

## 17.4.3. File Daemon

Feedback

The file daemon is managed via the configuration file `/etc/bacula/bacula-fd.conf` and must be set up on all systems that are to be backed up.

In the *Director* section, the `Name` option should be set to the name of the `director` (see Section 17.4.1). A client password must be set for every system. In addition, the `FDAddress` option in the *FileDaemon* section should be set to the computer's IP address.

```
Director {
  Name = sec-dir
  Password = "client-password"
}
```

```
FileDaemon {
  Name = sec-fd
  FDport = 9102
  WorkingDirectory = /var/lib/bacula
  Pid Directory = /var/run/bacula
  Maximum Concurrent Jobs = 20
  FDAddress = 192.0.2.125
}
```

Every computer to be backed up must also be registered in the `/etc/bacula/bacula-dir.conf` file in the director with the password specified above:

```
Client {
  Name = client-host
  Address = 192.0.2.125
  FDPort = 9102
  Catalog = MyCatalog
  Password = "client-password"
  File Retention = 30 days
  Job Retention = 6 months
  AutoPrune = yes
}
```

### 17.4.4. Bacula Console

Feedback

The Bacula console is managed in the `/etc/bacula/bconsole.conf` configuration file.

Here, the address of the computer on which the director daemon is running and its password must be entered in the *Director* section (see Section 17.4.1):

```
Name = localhost-dir
  DIRport = 9101
  address = 192.0.2.125
  Password = "master-dir-password"
```

### 17.4.5. Firewall adjustments

Feedback

In the basic Univention Firewall setting, the incoming packages are blocked/refused for all ports.

The ports used for Bacula must be approved accordingly. Access to the file daemon must be permitted on all systems. This is done by setting the Univention Configuration Registry variable `security/packetfilter/package/bacula/tcp/9102/all` to `ACCEPT` and then restarting Univention Firewall.

Port 9103 must also be approved in the same way on the Bacula server.

In a distributed setup, it may be necessary to permit the ports 9101/TCP (connections from the console to the directory) and 9103/TCP (connections from the directory and file daemon to the storage daemon) as well.

## 17.5. Configuration of the backup (interval, data, etc.)

Feedback

In Bacula one can define *resources* which when combined in a *job* represent a certain action, such as the backup of X data from Y computer on the Z medium. Among others, the following resources are available:

- Access to physical backup media is defined in a *device*, e.g., the type of device and how it is connected.
- The different backup media (e.g., tapes or hard drives) are identified as *volumes*. Volumes can be created manually or directly by the director. Bacula furnishes the volumes with software labels for identification.
- Bacula manages the volumes in *pools*. Any number of volumes can be combined and their properties defined. Backups are only performed for pools. When doing so, Bacula manages the utilization of the volumes and monitors when volumes can be overwritten again.
- A *schedule* defines when an action is performed. Additional options for an action can also be set or overwritten here.
- A *FileSet* defines which files or directories should be backed up, whether they should be compressed and which meta information (e.g., ACLs) should be backed up.
- Every computer from which the data should be backed up is treated as a *client* in Bacula. *Client* jobs define which computer is referred to and how the *file daemon* of the client can be accessed (e.g., password).

A *job* combines all of the information above. There are two types of job: restore and backup. In addition, the backup process of the backups (incremental, complete or differential backing up) is also defined here.

*Messages* are used to define how to handle Bacula status messages. Messages can be saved in log files, displayed on the console or sent by e-mail, for example.

[bacula-config-example] includes an example configuration which can be used as a template for backups and described the resources outline above in more detail.

## 17.6. Administration via the Bacula console

Feedback

The *Bacula console* can be used to export information about the status of Bacula, start backup jobs or restore data. It is started with the `bconsole` command.

The `status` command displays status information. A list of the director's upcoming, running and ended jobs is exported.

Backup jobs can be started automatically, e.g., every weekday. Backups and restores can also be started interactively via the Bacula console:

- The `run` command can be used to start a job. In addition, a list of available jobs is displayed from which one has to select the required job. The `mod` command can be used to set and change options such as the backup type for the job. Confirming with **`yes`** starts the job.
- The `restore` command can be used to restore data. `3 (Enter list of comma separated JobIds to select)` can now be used to select a backup job from which the data should be restored. A file browser then opens in which one can browse using the standard commands `cd` and `ls`. `mark FILE` and `mark -r DIR` select files and directories respectively for the restore. Once all the required data are selected, the file browser is exited with `done`. Once the client is specified and some options for the restore job confirmed (e.g., where the data should be copied to), the job can be started with **`yes`**. The selected data will be saved in the configured restore directory. If a tape is required for a backup or restore and it is not in the drive, Bacula requests the tape explicitly

Further information on the Bacula console can be found in the Bacula documentation or via the `help` command.

# 17.7. Backup of the catalog database

Feedback

The meta data of the backup are stored in the catalog. As standard, the catalog is stored in a PostgreSQL database, which should also be backed up. This is performed via a backup job, which saves an SQL dump of the database.

```
# Backup the catalog database (after the nightly save)
Job {
  Name = "BackupCatalog"
  JobDefs = "DefaultJob"
  Level = Full
  FileSet="Catalog"
  Schedule = "WeeklyCycleAfterBackup"
  # This creates an ASCII copy of the catalog
  # Arguments to make_catalog_backup.pl are:
  #  make_catalog_backup.pl catalog-name
  RunBeforeJob = "/etc/bacula/scripts/make_catalog_backup.pl MyCatalog"
  # This deletes the copy of the catalog
  RunAfterJob = "/etc/bacula/scripts/delete_catalog_backup"
  Write Bootstrap = "/var/lib/bacula/%n.bsr"
  Priority = 11
}

...

# This schedule does the catalog. It starts after the WeeklyCycle
Schedule {
  Name = "WeeklyCycleAfterBackup"
  Run = Full sun-sat at 23:10
}

...

# This is the backup of the catalog
FileSet {
  Name = "Catalog"
  Include {
    Options {
      signature = MD5
    }
    File = "/var/lib/bacula/bacula.sql"
  }
}
```

The instructions `RunBeforeJob` and `RunAfterJob` are run before and after the actual backing up of the scripts respectively. In the case of the catalog, `make_catalog_backup` is used prior to the backup to create an SQL dump of the catalog database and saved under `/var/lib/bacula/bacula.sql`. This file is then deleted again following successful backup.

In addition, `Write Bootstrap` is used to generate a bootstrap file for the backup of the catalog. This file documents how the data can be restored, i.e., on which volume they are saved and where on the volume they are. This is normally performed by the catalog itself, but for the backup of the catalog itself, a bootstrap file is required. It should also be backed up independently of Bacula.

The backup job for the catalog with the corresponding *FileSet* and *Schedule* is available as a template in the configuration of the *director daemon* and merely needs to be adjusted.

## 17.8. Further information

Feedback

Further information on the setup of Bacula is available on the following websites:

- http://www.bacula.org/
- http://wiki.bacula.org/doku.php
- http://www.bacula.org/5.2.x-manuals/en/main/main.pdf
- https://en.wikipedia.org/wiki/Bacula

# Bibliography


[ucs-dokumentationen] Univention GmbH. 2019. *UCS documentation overview*. https://docs.software-univention.de/.

[admx-reference] Microsoft. 2014. *Group Policy ADMX Syntax Reference Guide*. https://technet.microsoft.com/en-us/library/1db6fd85-d682-4d7d-9223-6b8dfafddc1c.

[admx-central] Mark Morowczynski. 2011. *How to Implement the Central Store for Group Policy Admin Templates, Completely (Hint: Remove Those .ADM files!)*. https://blogs.technet.microsoft.com/askpfeplat/2011/12/12/how-to-implement-the-central-store-for-group-policy-admin-templates-completely-hint-remove-those-adm-files/.

[microsoft-wmi-filter] Microsoft. 2005. *WMI filtering using GPMC*. https://www.microsoft.com/en-US/download/details.aspx?id=53314.

[add-wmi-filters] Mark Heitbrink. 2013. *Filtern von Gruppenrichtlinien anhand von Benutzergruppen, WMI und Zielgruppenadressierung*. http://www.gruppenrichtlinien.de/artikel/filtern-von-gruppenrichtlinien-anhand-von-benutzergruppen-wmi-und-zielgruppenadressierung/.

[adm-templates-howto] Florian Frommherz. 2007. *How to create custom ADM templates*. http://www.frickelsoft.net/blog/downloads/howto_admTemplates.pdf.

[microsoft-adm-templates] Microsoft. 2014. *Writing Custom ADM Files for System Policy Editor*. https://support.microsoft.com/en-us/kb/225087.

[bonding] Thomas Davis et al.. 2011. *Linux Ethernet Bonding Driver HOWTO*. https://www.kernel.org/doc/Documentation/networking/bonding.txt.

[dhcp-failover] ISC. 2013. *A Basic Guide to Configuring DHCP Failover*. https://kb.isc.org/article/AA-00502/31.

[developer-reference] Univention GmbH. 2022. *Univention Developer Reference*. https://docs.software-univention.de/developer-reference-5.0.html.

[release-notes] Univention GmbH. 2022. *UCS 5.0-2 Release Notes*. https://docs.software-univention.de/release-notes-5.0-2-en.html.

[bind-loglevel] O'Reilly. 1998. *Reading Bind Debugging Output*. http://www.diablotin.com/librairie/networking/dns-bind/ch12_01.htm.

[samba3-howto-chapter-20] Jelmer R. Vernooij and John H. Terpstra and Gerald (Jerry) Carter. 2010. *The Official Samba 3.2.x HOWTO and Reference Guide*. http://www.samba.org/samba/docs/Samba3-HOWTO.pdf#chapter.20.

[packaging-acl-extensions] Univention GmbH. 2022. *Packaging LDAP ACL Extensions*. https://docs.software-univention.de/developer-reference-5.0.html#settings:ldapacl.

[packaging-schema-extensions] Univention GmbH. 2022. *Packaging LDAP Schema Extensions*. https://docs.software-univention.de/developer-reference-5.0.html#settings:ldapschema.

[ucs-performance-guide] Univention GmbH. 2022. *UCS performance guide*. https://docs.software-univention.de/performance-guide-5.0.html.

[ext-doc-inst] Univention GmbH. 2022. *Extended installation documentation*. https://docs.software-univention.de/ext-installation/5.0/en/.

[ext-doc-uvmm] Univention GmbH. 2019. *Extended virtualization documentation*. https://docs.software-univention.de/uvmm-4.4.html.

[ext-doc-win] Univention GmbH. 2022. *Extended Windows integration documentation*. https://docs.software-univention.de/ext-windows/5.0/en/.

[ext-print-doc] Univention GmbH. 2022. *Extended print services documentation*. https://docs.software-univention.de/printers-4.4.html.

[ext-doc-domain] Univention GmbH. 2022. *Extended domain services documentation*. https://docs.software-univention.de/ext-domain/5.0/en/.

[ext-doc-net] Univention GmbH. 2022. *Extended IP and network management documentation*. https://docs.software-univention.de/ext-networks/5.0/en/.

[ec2-quickstart] Univention GmbH. 2016. *Univention Wiki - Amazon EC2 Quickstart*. http://wiki.univention.de/index.php?title=Amazon_EC2_Quickstart.

[xenserver-installation] Univention GmbH. 2016. *Univention Wiki - Citrix XenServer*. http://wiki.univention.de/index.php?title=Citrix_Xen_Server.

[bacula-config-example] Univention GmbH. 2013. *Bacula configuration example*. http://wiki.univention.de/index.php?title=Bacula_configuration_example.

[dovecot-wiki-clusterfs] Timo Sirainen. 2013. *Dovecot Wiki: Mail storage on shared disks*. http://wiki2.dovecot.org/MailLocation/SharedDisk.

[dovecot-wiki-nfs] Timo Sirainen. 2016. *Dovecot Wiki: NFS*. http://wiki2.dovecot.org/NFS.

[dovecot-wiki-services] Timo Sirainen. 2016. *Dovecot Wiki: Service configuration*. http://wiki2.dovecot.org/Services.