

Univention Corporate Server



Manual for users and administrators

Version 3.1-1
Date: March 25, 2013

Alle Rechte vorbehalten./ All rights reserved.

(c) 2002-2013
Univention GmbH
Mary-Somerville-Straße 1
28359 Bremen
Deutschland
feedback@univention.de

Jede aufgeführte Marke und jedes Warenzeichen steht im Eigentum ihrer jeweiligen eingetragenen Rechteinhaber. 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.

Table of Contents

1. Introduction	11
1.1. What is Univention Corporate Server?	11
1.2. Overview of UCS	12
1.2.1. Commissioning	12
1.2.2. Domain concept	12
1.2.3. Expandability with components	13
1.2.4. LDAP directory service	13
1.2.5. Domain administration	14
1.2.6. Computer administration	15
1.2.7. Policy concept	15
1.2.8. Listener/notifier replication	15
1.2.9. Virtualization management	16
1.3. Further documentation	16
1.4. Symbols and conventions used in this manual	16
2. Installation	19
2.1. Introduction	19
2.1.1. Operating the Installer	19
2.2. Selecting the installation mode	20
2.3. Selecting the system language	21
2.4. Selecting the installation medium	22
2.5. Selecting the time zone	22
2.6. Selecting the keyboard layout	22
2.7. Selecting the system role	23
2.8. Domain settings	24
2.9. Hard drive partitioning	25
2.10. Network configuration	28
2.10.1. Configuration of IPv4 addresses	28
2.10.2. Configuration of IPv6 addresses	28
2.11. Join options	29
2.12. Selecting software components	30
2.13. Installation overview	31
2.14. Troubleshooting for installation problems	33
2.15. Installation in the Amazon EC2 cloud	33
2.16. Installation in Citrix XenServer	33
3. Domain services / LDAP directory	35
3.1. Introduction	36
3.2. Joining domains	36
3.2.1. How UCS systems join domains	36
3.2.1.1. Subsequent domain joins with univention-join	37
3.2.1.2. Joining domains with Univention Management Console	37
3.2.1.3. Subsequent running of join scripts	37
3.2.2. How to join domains with Windows clients	37
3.2.2.1. Windows 8	38
3.2.2.2. Windows 7	39
3.2.2.3. Windows XP Professional	39
3.2.3. Ubuntu domain joins	39
3.2.4. Mac OS X domain joins	39
3.2.4.1. Domain join using the system preferences GUI	40
3.2.4.2. Domain join on the command line	40
3.3. UCS system roles	41
3.3.1. Domain controller master	41
3.3.2. Domain controller backup	41

3.3.3. Domain controller slave	41
3.3.4. member server	41
3.3.5. Base system	41
3.3.6. Ubuntu	41
3.3.7. Linux	41
3.3.8. Univention Corporate Client	41
3.3.9. Mac OS X	42
3.3.10. Managed client	42
3.3.11. Mobile client	42
3.3.12. Thin Client	42
3.3.13. Domain Trust Account	42
3.3.14. IP managed client	42
3.3.15. Windows Domaincontroller	42
3.3.16. Windows Workstation/Server	42
3.4. LDAP directory	42
3.4.1. LDAP schemas	42
3.4.1.1. LDAP schema extensions	43
3.4.1.2. LDAP schema replication	43
3.4.2. Audit proof logging of LDAP changes	43
3.4.3. Timeout for inactive LDAP connections	43
3.4.4. LDAP command line tools	43
3.4.5. Access control for the LDAP directory	44
3.4.5.1. Delegation of the privilege to reset user passwords	44
3.4.6. Name Service Switch / LDAP NSS module	45
3.4.7. Syncrep for synchronisation with non-UCS OpenLDAP servers	45
3.4.8. Configuration of the directory service when using Samba 4	45
3.5. Domain replication using the listener/notifier system	45
3.6. SSL certificate management	46
3.7. Kerberos	47
3.8. Converting a DC backup to the new DC master	48
4. Central domain management with UMC	49
4.1. Introduction	49
4.2. Operating instructions for Univention Management Console	50
4.2.1. Login	50
4.2.2. Activation of UCS license / license overview	51
4.2.3. Operating instructions for domain modules	51
4.2.3.1. Searching for objects	52
4.2.3.2. Creating objects	53
4.2.3.3. Editing objects	53
4.2.3.4. Deleting objects	53
4.2.3.5. Moving objects	53
4.2.4. Favorites	53
4.2.5. Feedback on UMC and UCS	54
4.3. Collection of usage statistics	54
4.4. LDAP directory browser	54
4.5. Policies	55
4.5.1. Creating a policy	55
4.5.2. Applying policies	56
4.5.3. Editing a policy	56
4.6. Expansion of the attributes managed in UMC with extended attributes	56
4.7. Structuring of the domain with user-defined LDAP structures	60
4.8. Command line interface of domain management (Univention Directory Manager)	61
4.9. Evaluation of data from the LDAP directory with Univention Directory Reports	63
4.9.1. Creating reports in the Univention Management Console	64

4.9.2. Creating reports on the command line	64
4.9.3. Adjustment/expansion of Univention Directory Reports	64
5. User management	65
5.1. User management in the Univention Management Console	65
5.2. User password management	72
5.3. Password settings for Windows clients when using Samba	73
5.4. Password changes by users via Univention Management Console	74
5.5. Automatic lockout of users after failed login attempts	74
5.6. User templates	75
6. Group management	77
6.1. Managing groups in the Univention Management Console	77
6.2. Nested groups	79
6.3. Local group cache	79
7. Computer management	81
7.1. Management of computer accounts in the Univention Management Console	82
7.1.1. Integration of Ubuntu clients	85
7.2. Configuration of hardware and drivers	85
7.2.1. Available kernel variants	85
7.2.2. Hardware drivers / kernel modules	86
7.2.3. GRUB boot manager	86
7.2.4. Network configuration	87
7.2.4.1. Network interfaces	87
7.2.4.2. Configuring proxy access	89
7.2.5. Configuration of the monitor settings	89
7.2.6. Mounting NFS shares	90
7.2.7. Collection of list of supported hardware	90
7.3. Administration of local system configuration with Univention Configuration Registry	90
7.3.1. Introduction	90
7.3.2. Using the Univention Management Console web interface	92
7.3.3. Using the command line front end	92
7.3.3.1. Querying a UCR variable	92
7.3.4. Policy-based configuration of UCR variables	94
7.3.5. Modifying UCR templates	94
7.3.5.1. Referencing of UCR variables in templates	95
7.3.5.2. Integration of inline Python code in templates	95
7.4. Basic system services	95
7.4.1. Administrative access with the root account	96
7.4.2. Configuration of language and keyboard settings	96
7.4.3. Starting/stopping system services / configuration of automatic startup	96
7.4.4. Authentication / PAM	97
7.4.4.1. Limiting authentication to selected users	97
7.4.5. Configuration of the LDAP server in use	98
7.4.6. Configuration of the print server in use	98
7.4.7. Logging/retrieval of system messages and system status	98
7.4.7.1. Log files	98
7.4.7.2. Logging the system status	98
7.4.7.3. Querying system statistics in the Univention Management Console	99
7.4.7.4. Process overview in the Univention Management Console	99
7.4.8. Executing recurring actions with Cron	100
7.4.8.1. Hourly/daily/weekly/monthly execution of scripts	100
7.4.8.2. Defining local cron jobs in /etc/cron.d	100
7.4.8.3. Defining cron jobs in Univention Configuration Registry	100
7.4.9. Nameserver cache daemon	101
7.4.10. SSH login to systems	101

7.4.11. Configuring the time zone / time synchronisation	102
7.5. Software deployment	102
7.5.1. Introduction	102
7.5.2. Differentiation of update variants / UCS versions	103
7.5.3. Univention App Center	103
7.5.4. Configuration of the repository server for updates and package installations	105
7.5.4.1. Configuration via the Univention Management Console	106
7.5.4.2. Configuration via Univention Configuration Registry	106
7.5.4.3. Policy-based configuration of the repository server	106
7.5.4.4. Creating and updating a local repository	106
7.5.5. Updates of UCS systems	107
7.5.5.1. Update strategy in environments with more than one UCS system	107
7.5.5.2. Updating individual systems via the Univention Management Console	107
7.5.5.3. Updating individual systems via the command line	108
7.5.5.4. Updating systems via a policy	109
7.5.5.5. Postprocessing of release updates	109
7.5.5.6. Troubleshooting in case of update problems	109
7.5.6. Installation of further software	109
7.5.6.1. Role-based software selection in the Univention Management Console	110
7.5.6.2. Installation/removal of individual packages in the Univention Management Console	110
7.5.6.3. Installation/removal of individual packages in the command line	111
7.5.6.4. Policy-based installation/uninstallation of individual packages via package lists	111
7.5.7. Specification of an update point using the package maintenance policy	112
7.5.8. Integration of components not provided through the Univention App Center	112
7.5.8.1. Integration of repository components via the Univention Management Console	113
7.5.8.2. Integration of repository components via Univention Configuration Registry	114
7.5.9. Central monitoring of software installation statuses with the software monitor	114
8. Services for Windows	117
8.1. Introduction	117
8.2. Installation and components of a Samba domain	118
8.2.1. Installation and components of a Samba 3 domain	118
8.2.2. Installation and components of a Samba 4 domain	119
8.3. Services of a Samba domain	119
8.3.1. Authentication services	119
8.3.2. File services	119
8.3.3. Print services	120
8.3.4. Univention S4 connector	120
8.3.5. NetBIOS name service	120
8.3.6. Name resolution using WINS	120
8.4. Configuration and management of Windows desktops	121
8.4.1. Group policies	121
8.4.2. Logon scripts / NETLOGON share	121
8.4.2.1. Samba 3	121
8.4.2.2. Samba 4	122
8.4.3. Configuration of the file server for the home directory	122
8.4.3.1. Configuration with Samba 3	122
8.4.3.2. Configuration with Samba 4	122
8.4.4. Roaming profiles	122
8.4.4.1. Samba 3	122
8.4.4.2. Samba 4	123

8.4.5. Granting additional Windows privileges to users	123
8.5. UCS Active Directory Connector	123
8.5.1. Introduction	123
8.5.2. Setup of the UCS AD connector	124
8.5.2.1. Basic configuration of the connector	124
8.5.2.2. Importing the SSL certificate of the Active Directory	126
8.5.2.3. Setting up the password service on the AD system	127
8.5.2.4. Starting/Stopping the Active Directory connector	128
8.5.2.5. Functional test of basic settings	128
8.5.3. Additional tools / Debugging connector problems	128
8.5.3.1. univention-adsearch	128
8.5.3.2. univention-connector-list-rejected	128
8.5.3.3. Logfiles	129
8.5.4. Details on preconfigured synchronisation	129
8.5.4.1. Containers and organisational units	129
8.5.4.2. Groups	129
8.5.4.3. Users	130
8.6. Trust relationships	131
8.6.1. Windows domain trusts Samba 3 domain	131
8.6.2. Samba 3 domain trusts Windows domain	131
9. IP and network management	133
9.1. Network objects	134
9.2. Administration of DNS data with Bind	135
9.2.1. Configuration of the Bind nameserver	136
9.2.1.1. Configuration of Bind debug output	136
9.2.1.2. Configuration of the data backend	136
9.2.1.3. Configuration of zone transfers	136
9.2.2. Administration of DNS data in the Univention Management Console	137
9.2.2.1. Forward lookup zone	137
9.2.2.2. CNAME-Record (Alias-Records)	139
9.2.2.3. A/AAAA records (host records)	139
9.2.2.4. Service records	139
9.2.2.5. Reverse lookup zone	141
9.2.2.6. Pointer record	141
9.2.3. IP assignment via DHCP	142
9.3.1. Introduction	142
9.3.2. Composition of the DHCP configuration via DHCP LDAP objects	143
9.3.2.1. Administration of DHCP services	143
9.3.2.2. Administration of DHCP server entries	143
9.3.2.3. Administration of DHCP subnets	143
9.3.2.4. Administration of DHCP pools	144
9.3.2.5. Registration of computers with DHCP computer objects	145
9.3.2.6. Management of DHCP shared networks / DHCP shared subnets	145
9.3.3. Configuration of clients via DHCP policies	146
9.3.3.1. Setting the gateway	146
9.3.3.2. Setting the DNS servers	146
9.3.3.3. Setting the WINS server	146
9.3.3.4. Configuration of the DHCP lease	147
9.3.3.5. Configuration of boot server/PXE settings	147
9.3.3.6. Further DHCP policies	148
9.4. Packet filter with Univention Firewall	148
9.5. Web proxy for caching and policy management / virus scan	148
9.5.1. Installation	149
9.5.2. Caching of web content	149

9.5.3. Logging proxy accesses	149
9.5.4. Restriction of access to permitted networks	149
9.5.5. Configuration of the ports used	150
9.5.5.1. Access port	150
9.5.5.2. Permitted ports	150
9.5.6. User authentication on the proxy	150
9.5.7. Filtering/policy enforcement of web content with Dansguardian	151
9.5.8. Definition of content filters for Dansguardian	152
10. Fileshare management	155
10.1. Access rights to data in shares	155
10.2. Management of shares in the Univention Management Console	156
10.3. Support for MSDFS	162
11. Print services	165
11.1. Introduction	165
11.2. Installing a print server	165
11.3. Setting the local configuration properties of a print server	166
11.4. Creating a printer share	166
11.5. Creating a printer group	169
11.6. Administration of print jobs and print queues	170
11.7. Generating PDF documents from print jobs	171
11.8. Mounting of print shares in Windows clients	171
11.9. Integrating additional PPD files	172
12. Mail services	173
12.1. Introduction	173
12.2. Installation	174
12.3. Management of the mail server data	174
12.3.1. Management of mail domains	174
12.3.2. Assignment of e-mail addresses to users	175
12.3.3. Management of mailing lists	175
12.3.4. Management of mail groups	176
12.3.5. Management of shared IMAP folders	177
12.3.6. Mail quota	178
12.4. Spam detection and filtering	179
12.5. Identification of viruses and malware	180
12.6. Integration of Fetchmail for retrieving mail from external mailboxes	180
12.7. Configuration of the mail server	181
12.7.1. Configuration of a relay host for sending the e-mails	181
12.7.2. Configuration of the maximum mail size	181
12.7.3. Configuration of a blind carbon copy for mail archiving solutions	181
12.7.4. Configuration of soft bounces	181
12.7.5. Handling of mailboxes during e-mail changes and the deletion of user accounts	182
12.7.6. Distribution of an installation on several mail servers	182
12.8. Webmail and administration of e-mail filters with Horde	182
12.8.1. Login and overview	182
12.8.2. Web-based mail access	183
12.8.3. Address book	183
12.8.4. E-mail filters	184
13. Infrastructure monitoring with Nagios	187
13.1. Introduction and structure	187
13.2. Installation	188
13.2.1. Preconfigured Nagios checks	189
13.3. Configuration of the Nagios monitoring	191
13.3.1. Configuration of a Nagios service	191
13.3.2. Configuration of a monitoring time period	193

13.3.3. Assignment of Nagios checks to computers	194
13.3.4. Einbindung von manuell erstellten Konfigurationsdateien	196
13.4. Querying the system status via the Nagios web interface	196
14. Virtualization	197
14.1. Introduction	197
14.2. Installation	198
14.3. Image files of virtual machines	198
14.4. Accessing the default storage pool through a file share	199
14.5. CD/DVD/floppy drives in virtual machines	200
14.6. Network interfaces in virtual instances	200
14.7. Paravirtualization/virtIO drivers for Microsoft Windows systems	201
14.7.1. Installation of the GPLPV drivers for Xen instances	201
14.7.2. Installation of the virtIO drivers for KVM instances	202
14.8. Snapshots	202
14.9. Migration of virtual instances	203
14.9.1. Migration of virtual machines from failed virtualization servers	203
14.10. Managing virtual machines with the Univention Management Console	203
14.10.1. Operations (Starting/stopping/suspending/deleting/migrating/cloning virtual machines)	204
14.10.2. Creating a virtual instance	205
14.10.3. Modifying virtual machines	206
15. Data backup with Bacula	209
15.1. Introduction	209
15.2. Installation	210
15.3. Configuration of the backup components	210
15.3.1. Directory Daemon	210
15.3.2. Storage	211
15.3.3. File Daemon	211
15.3.4. Bacula Console	212
15.3.5. Firewall adjustments	212
15.4. Configuration of the backup (interval, data, etc.)	212
15.5. Administration via the Bacula console	213
15.6. Backup of the catalog database	213
15.7. Further information	214
Bibliography	215

Chapter 1. Introduction

1.1. What is Univention Corporate Server?	11
1.2. Overview of UCS	12
1.2.1. Commissioning	12
1.2.2. Domain concept	12
1.2.3. Expandability with components	13
1.2.4. LDAP directory service	13
1.2.5. Domain administration	14
1.2.6. Computer administration	15
1.2.7. Policy concept	15
1.2.8. Listener/notifier replication	15
1.2.9. Virtualization management	16
1.3. Further documentation	16
1.4. Symbols and conventions used in this manual	16

1.1. What is Univention Corporate Server?

[Feedback](#) 

Univention Corporate Server (UCS) is a Linux-based corporate platform 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 includes and is based on reliable Open Source software tried and tested in organisations 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 and virtualization management system
- Support for Microsoft Windows and integration in Microsoft Active Directory environments
- 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. UCS basic system
2. UCS management system
3. UCS components

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

The *UCS management system* realises 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 or in the command line and in individual scripts. The UCS management system can be extended with APIs (application programmers interfaces) and provides a flexible client-server architecture which allows changes to be transferred to the involved systems and be activated there.

Additional *UCS components* from Univention and other manufacturers 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](#) 

As an operating system designed for multi-user and multi-tasking application right from the start, where the focus in its development was always on stability, security and compatibility with other operating systems, Linux is predestined for being used in server operating systems that are stable, secure and available at all times. Without UCS, however, the administration of Linux systems is often time-consuming, complicated and largely unstandardised, particularly in application scenarios in which infrastructure services closely linked to user and computer administration are provided for companies and authorities. This is the point where UCS comes into play.

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 utilisation.

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, which is also set up via an installation or can be generated from an existing virtual machine employed as a template. The installation can be interactive or fully automatic either from a CD-ROM/DVD or via the network/PXE.

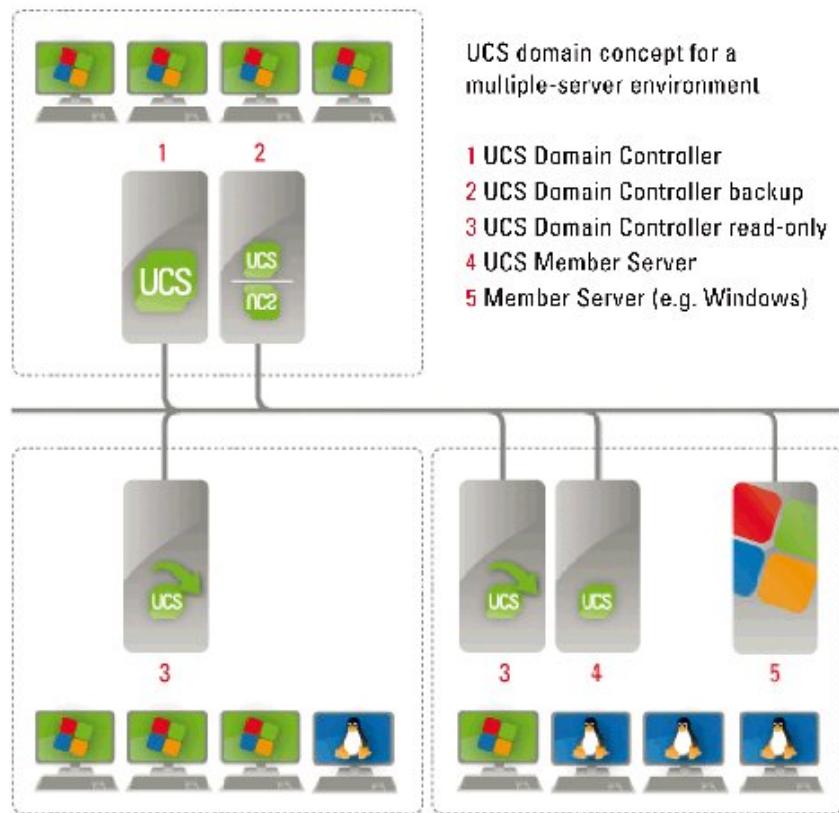
Further information can be found in Chapter 2.

1.2.2. Domain concept

[Feedback](#) 

During installation each computer is assigned a system role. As standard in UCS, all servers, clients and users are stored in one security and trust context, the so-called UCS domain. Consequently, all the system roles (domain controllers, member servers and clients) are available.

Figure 1.1. UCS domain concept



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

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 and are available in the Samba 4 domain via UCS.

Ubuntu clients can be integrated in a UCS environment (see Section 7.1.1).

[Feedback](#)

1.2.3. Expandability with components

The functional scope of a computer can be expanded via the selection of different additional components during the installation or via an online repository. The components can be setup without time-consuming configuration work as they are coordinated for UCS's holistic concept and integrate seamlessly into the UCS management system.

[Feedback](#)

1.2.4. LDAP directory service

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

[Feedback](#)

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. 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 and users can log in there with their domain passport.

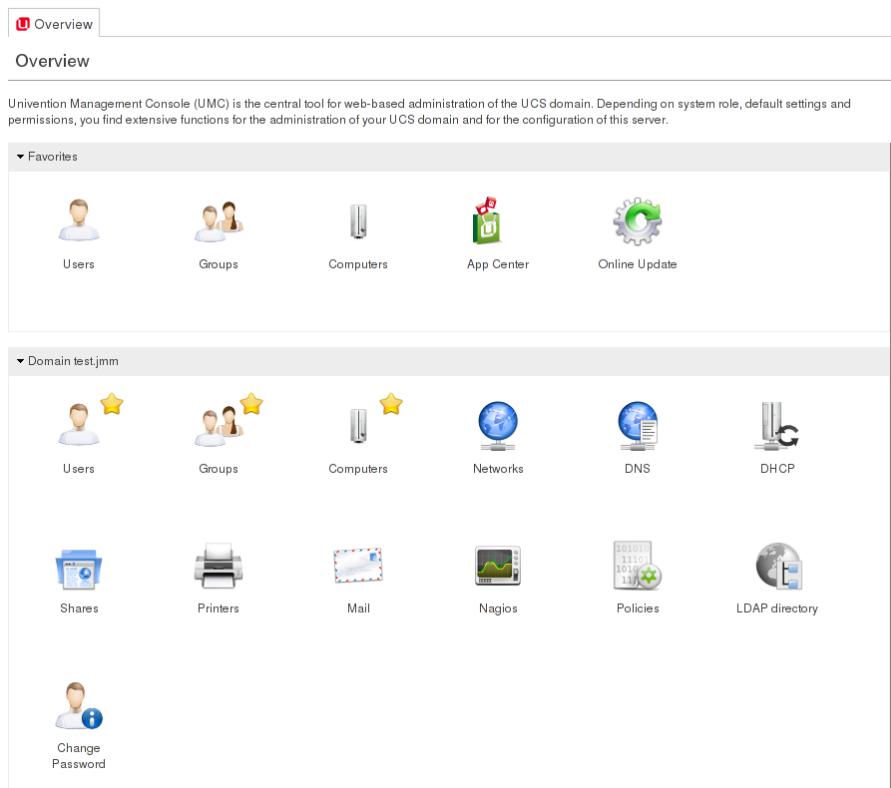
UCS utilises OpenLDAP as a directory service server. The directory is provided by the master domain controller and replicated on all domain controllers (DCs) in the domain. The complete LDAP directory is also replicated on a DC backup as this can replace the DC master in an emergency. In contrast, the replication on DC slaves can be restricted to certain areas of the LDAP directory using ACLs (access control lists), which allows selective replication. For example, this may be desirable if data should only be stored on as few servers as possible for security reasons. A root CA (certification authority) is integrated in UCS so that these and other data can be transmitted encrypted.

Further information can be found in Section 3.4.

[Feedback](#)

1.2.5. Domain administration

Figure 1.2. Univention Management Console



Access to the LDAP directory is performed via the web-based and modular Univention Management Console (UMC). Technically, the access is performed by the Univention Management Console module Univention Directory Manager. This is displayed as the **Domain** modules in Univention Management Console.

Univention Directory Manager also allows the realisation 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 makes it possible to enter data into the LDAP directory and then display, edit and delete them there. It is also possible to search - filtered according to a wide range of criteria - for data. 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 and attributes.

Further information can be found in Chapter 4.

[Feedback](#) 

1.2.6. Computer administration

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

[Feedback](#) 

1.2.7. Policy concept

The tree-like structure of LDAP directories ensures - similar to in a file system - that all objects (such as users, computers, etc.) are in one container and that these containers can also be adopted by other containers. The so-called root container forms the basis for this and is often also called the LDAP base object.

Policies describe certain administrative settings which can be practically be used on more than one object. They facilitate the administration as they can be linked to containers and then apply to all the objects in the container in question and the objects in subfolders.

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

Further information can be found in Section 4.5.

[Feedback](#) 

1.2.8. Listener/notifier replication

The listener/notifier mechanism is an important technical component of the UCS management system. With this, the creation, editing or deleting of freely definable 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 - even customer-specific - procedures and is consequently 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 management

[Feedback](#) 

With the UMC module UCS Virtual Machine Manager (UVMM), UCS offers an extensive, powerful tool for the administration of virtualized infrastructures. Virtualization servers registered in the UCS domain and virtual machines operated on it can be centrally monitored and administrated. The Open Source virtualization technologies KVM (Kernel Virtual Machine) and Xen can be used as hypervisor technology with UCS.

Further information can be found in Chapter 14.

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 (e. g. automatic Windows client installation)
- Central network management with DNS and DHCP
- Monitoring systems and networks with Nagios
- Print server functionalities
- Thin Client support
- Fax service
- Proxy server
- Virtualization with Xen and KVM
- 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 (<http://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.

Attribute	Description
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 brackets.

Names are in *bold*.

Computer names, LDAP DNs, program names, file names, file paths, internet addresses und 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	19
2.1.1. Operating the Installer	19
2.2. Selecting the installation mode	20
2.3. Selecting the system language	21
2.4. Selecting the installation medium	22
2.5. Selecting the time zone	22
2.6. Selecting the keyboard layout	22
2.7. Selecting the system role	23
2.8. Domain settings	24
2.9. Hard drive partitioning	25
2.10. Network configuration	28
2.10.1. Configuration of IPv4 addresses	28
2.10.2. Configuration of IPv6 addresses	28
2.11. Join options	29
2.12. Selecting software components	30
2.13. Installation overview	31
2.14. Troubleshooting for installation problems	33
2.15. Installation in the Amazon EC2 cloud	33
2.16. Installation in Citrix XenServer	33

2.1. Introduction

[Feedback](#) 

Several types of installation are supported:

- Univention Corporate Server can be *installed from a DVD*. The installation of the first UCS system of a domain usually occurs from DVD. There is either a text-based installation or an automated installation via a pre-configured installation profile:
 - With the interactive installation, all system settings are to be entered by the user.
 - With profile-based installation, system settings are stored in text files, the so-called installation profiles. Further information on the format of the installation profile files can be found in the extended installation documentation [ext-doc-inst]. Installation profiles can be integrated locally via floppy disks or USB storage media, or they can be stored beforehand on the installation medium in the form of an adjusted installation DVD.
- The Univention Net Installer can be used to perform *network based installation* automatically via PXE. The settings are also taken from installation profiles. This is documented in the extended installation documentation [ext-doc-inst]

Aside from installation on hardware or in a virtualization solution, UCS can also be installed in the Amazon EC2 cloud using an AMI image. Further information can be found in Section 2.15.

2.1.1. Operating the Installer

[Feedback](#) 

The input screens of the Univention Installer are operated via keyboard exclusively:

- The **TAB** key moves the cursor to the next entry field.
- Use **Shift+TAB** to go back to the previous entry field.
- The **RETURN** key is used for transmitting values entered into an entry field, and for confirming buttons.
- Within a list or table, the *arrow keys* can be used for navigating between entries.

Selecting the installation mode

- For further information on a module, the **F1** function key may be pressed for calling a help dialog.
- The next module is called via the **F12** function key, while the **F11** function key lets you go back to modules already processed. As an alternative to **F11** and **F12**, the buttons can also be used.

Further assignments of function keys are given below in the descriptions of the individual modules.

Note

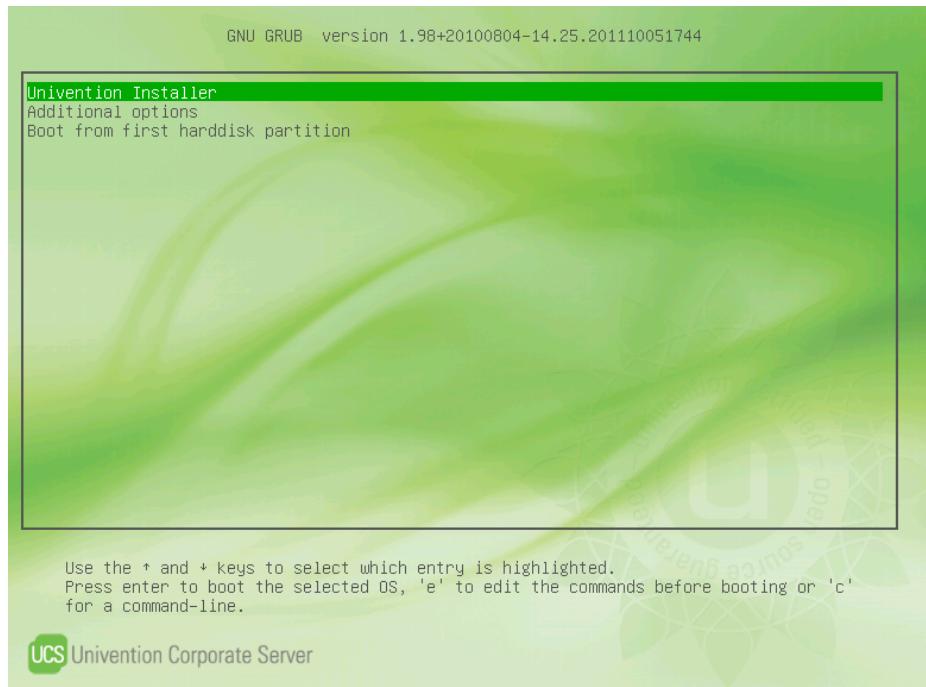
In the left-hand window on the installation surface, a list of all the modules is displayed. The active module is highlighted in colour. This list cannot be used for navigation, it is only for orientation purposes.

2.2. Selecting the installation mode

[Feedback](#) 

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

Figure 2.1. Selecting the boot prompt



Now you can choose between several installation procedures.

- *Univention Installer* starts the interactive installation routine on the basis of the Linux kernel in version 3.2. During installation, the installer requests a number of parameters such as network settings, hard drive partitions, system role, and the selection of software components for the computer to be installed. Then the installation is performed.
- *Additional options* allows the selection of advanced options for the installation process. First a request appears, requesting the Linux kernel to be installed. Different installation variants can then be selected:
 - **Univention Installer (normal mode)** performs an interactive standard installation.
 - **Univention Installer Software RAID (expert mode)** also starts a interactive installation like the previous mode. In this version the partitioning, however, is not performed with the support of the installer,

but effected manually. To do this a number of programs including `fdisk`, `mdadm` and `mkefs.ext3` are available. This mode can, for example, be used to set up a software RAID or an encrypted hard drive partition. After the partitioning, the interactive installation is continued. This is documented in the extended installation documentation [ext-doc-inst]

- **Univention Installer MBR-Mode (deprecated)** starts an interactive installation, during which the partitioning module writes a master boot record instead of following the GPT standard. Further details can be found in Section 2.9.
- **Univention Installer Profile** starts the profile-based installation. With profile-based installation, a pre-arranged installation profile is selected from which the system reads the installation parameters. The installation profile to be used is taken from a directory on the installation medium. Profile-based installation is described in the extended installation documentation [ext-doc-inst]
- **Univention Installer Profile Floppy** also starts a profile-based installation. The installation profile is taken from a floppy disk.
- **Univention Installer Profile USB** also starts a profile-based installation. The installation profile is taken from a USB storage medium.
- **Boot from first hard drive partition** does not start the UCS installation, but the operating system installed on the first hard drive instead.

Now, the kernel is loaded from the installation medium and status messages appear on the screen. The installation itself is divided into modules. Each module contains settings which are connected as regards content. There are modules for network configuration or for selecting the software to be installed, among others.

2.3. Selecting the system language

[Feedback](#)

Here 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.

Figure 2.2. Selecting the system language



Selecting the installation medium

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

The language which is to be used during the installation process, is selected here. Available languages are German and English.

2.4. Selecting the installation medium

[Feedback](#)

The installation medium is detected. The installation medium to be used is specified via the **F2** function key. This parameter can be used in case problems with one or several CD drives occur, or if the net-based installation is to be executed from a different installation server. The **F4** function key is used for rescanning the list of installation media.

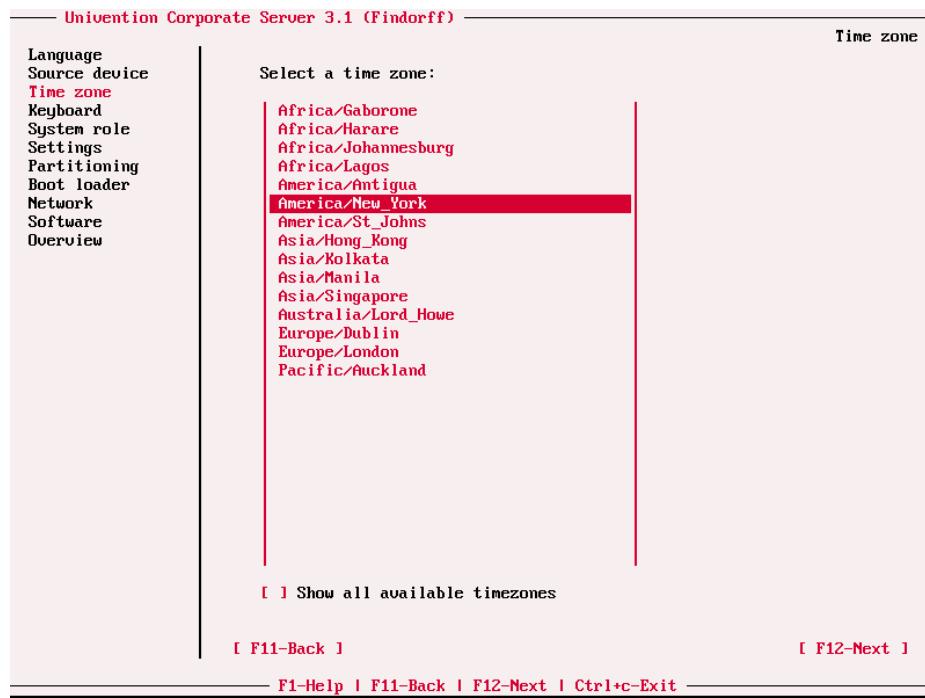
If the installation medium is detected automatically, the Univention Installer changes to the next module autonomously. The function key **F11** can be used to return this module at a later point.

2.5. Selecting the time zone

[Feedback](#)

Here the desired time zone can be selected. Depending on the selected system language, a selection of appropriate time zones is displayed. All the available time zones can be listed by activating the option **Show all available time zones**.

Figure 2.3. Time zone selection

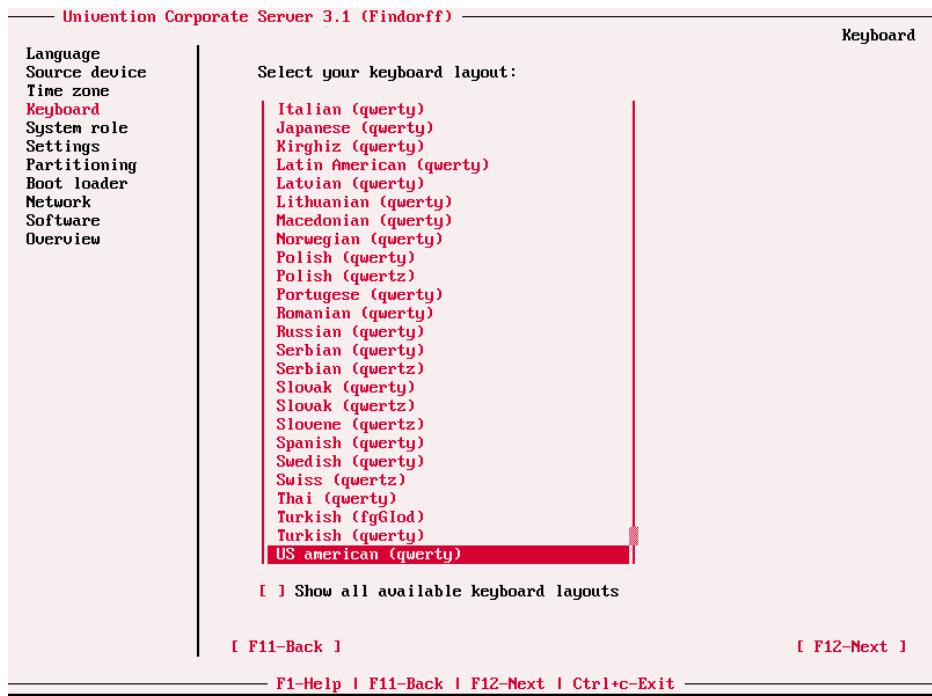


2.6. Selecting the keyboard layout

[Feedback](#)

Here the keyboard layout of the keyboard in use is selected. Depending on the selected system language, a selection of appropriate keyboard layouts is displayed. All the available keyboard versions can be listed by activating the option **Show all available keyboard layouts**.

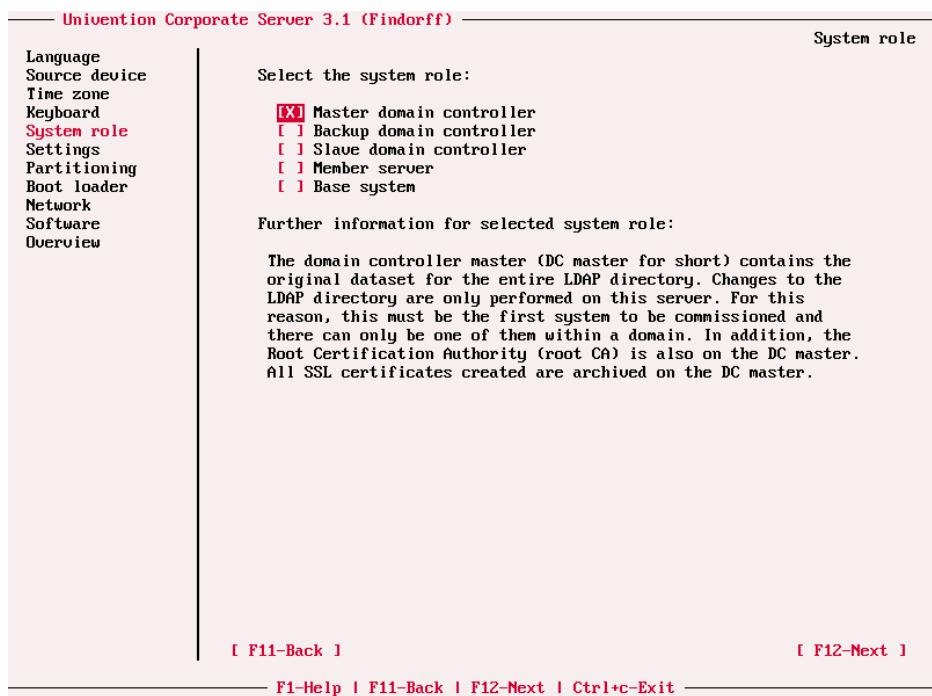
Figure 2.4. Keyboard layout selection



2.7. Selecting the system role

[Feedback](#)

Figure 2.5. System role selection



The system role for the system to be installed is selected here. The installation dialogue also includes a short description.

Domain settings

The following are available to choose from:

1. Master domain controller
2. Backup domain controller
3. Slave domain controller
4. Member server
5. Base system

The first system to be installed on a UCS domain should always be a master domain controller. The installation of further UCS systems requires a domain controller master to be running during installation. The sole exception to this is the *base system* system role, which can be installed without a connection to a master domain controller.

The properties of the system roles are described in Section 3.3. The installation dialogue also displays a short description.

2.8. Domain settings

[Feedback](#) 

The fully qualified domain name (FQDN) of the Univention Corporate Server system under which the computer should be accessible in the network should be entered under **Fully qualified domain name** (e.g., mail.example.com).

Host names must consist of a combination of the following characters: letters a-z in lower case, numerals 0-9, hyphens, and underlines. A host name has to begin with a letter and must not be longer than 15 characters.

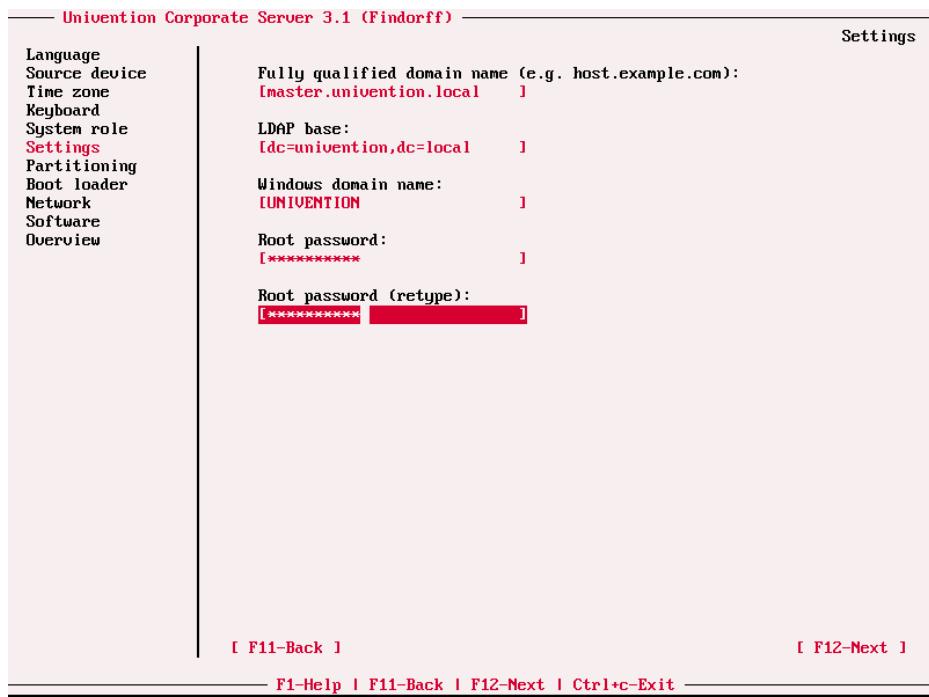
Once the domain name is set, the fields **LDAP base** and **Windows domain** will be assigned by derived values. If the LDAP basis is to be changed, the name conventions for DNs have to be observed. The basis DN has to begin with *cn=*, *dc=*, *c=*, *l=*, or *o=*. Basis DNs violating these conventions will be rejected.

The name of the Windows domain may have a maximum length of 15 characters; it may consists of letters, numerals and hyphens exclusively. The entry field **LDAP base** appears solely in the system role of master domain controller.

Root password is the password for the user *root*. If a master domain controller is installed, this password is also entered for the user *Administrator*. In subsequent operation, the passwords of the users *root* and *Administrator* can be managed independently. The password is entered for a second time in the final entry field. This double input is necessary to avoid spelling errors since the entered characters are not displayed on the screen.

For security reasons, the root passwords needs to consist of at least eight characters.

Figure 2.6. Domain settings



2.9. Hard drive partitioning

[Feedback](#)

This menu is used for partitioning the existing hard drives.

As of UCS 3.1, the Univention Installer performs a partitioning in accordance with the GUID Partition Table standard (GPT). Earlier UCS versions used a partitioning in accordance with the Master Boot Record standard, whereby the partitioning was stored in the first data block of a storage device.

The GPT is downwardly compatible with the MBR; the first data block is still stored in MBR format. This block is designated as the *Protective MBR* and contains a single protective partition. It prevents, for example, that older partitioning tools damage the GPT data or data within the partitions.

It is possible to create up to 127 partitions, which can take on sizes in the zettabyte range.

The *Unified Extensible Firmware Interface (UEFI)* replaces PC-BIOS on newer systems. As of Version 3.1, UCS supports a boot via UEFI in the 64 bit version (amd64).

On systems with a BIOS - in other words systems without UEFI - a BIOS boot partition is necessary when using GPT, in which the Grub boot loader can store the majority of its functions.

If the Univention Installer detects an existing master boot record on the system to be installed, this can be ignored (the disk is then not taken into account in the GPT partitioning) or the MBR partitions converted into GPT partitions.

As of UCS 3.1, the Univention Installer optimises the layout of the data blocks on the disk in order to achieve optimal performance on hard drives with 4k sectors or solid-state disks. If an existing master boot record is converted into a GPT, the data blocks are not reorganised and as such they are possibly not optimally organised. Consequently, preference should be given to a new partitioning.

The conversion to a GPT can affect other installed operating systems, especially if they or their bootloaders do not support GPT (e.g., Windows 2000, the 32-bit versions of Window XP or older Linux versions).

Hard drive partitioning

At the first opening of this menu, the installer offers automatic partitioning of the system. On confirmation, the installer displays a partitioning proposal in which all the existing hard drives are newly partitioned and formatted using the Logical Volume Manager (LVM).

Caution

All the data stored on these hard drives will be lost during this process! Should the proposed partitioning be undesirable, it can be rejected by pressing the **F5** function key. The disk space of external storage media, such as USB sticks and USB hard drives, is also included in the automatic partitioning procedure, so these media should be disconnected if necessary.

If the auto-partitioning is used, the following partitions are created on non-UEFI systems:

- an 8 MB BIOS boot partition
- a 512 MB /boot partition
- a swap partition with twice the current RAM (min. 512 MB; max. 10 GB)
- a / partition with the remaining disk space

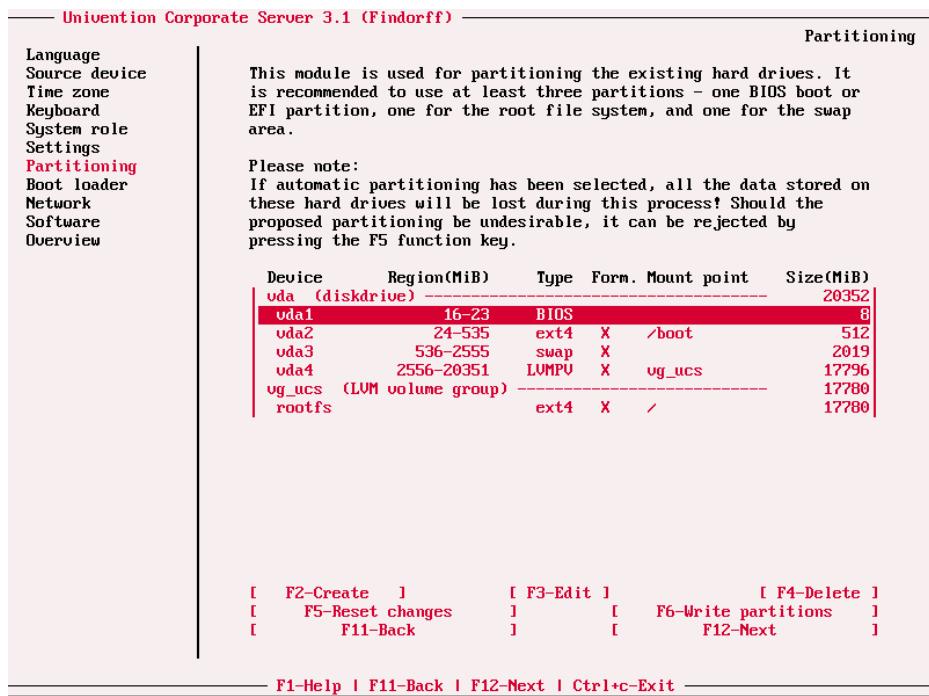
The following partitions are created on UEFI systems by the auto-partitioning:

- a 256 MB EFI system partition
- a 512 MB /boot partition
- a swap partition with twice the current RAM (min. 512 MB; max. 10 GB)
- a / partition with the disk drive space

The properties of the detected drives, existing partitions, and LVM media (name, range of a partition, type, formatting, directory in which the partition is to be mounted, and size in MB) are displayed in a table.

For creating a new partition, select the **free** entry of the desired drive and select the **[F2-Create]** button. In the next dialogue, enter the directory where the partition is to be mounted (mount point), the size of the partition to be created in megabytes, the file system which is to be created on the partition, and the type of partition (BIOS boot, EFI, data, swap, LVM). If a partition is to be created for the swap area, it is not necessary to give the name of a directory where the partition is to be mounted. If all settings are made, click the **[F6-Write partitions]** button to confirm and store the settings.

Figure 2.7. Partitioning hard drives



If you wish to include a partition from a former installation, or to correct settings made during the creation of a new partition, just select the appropriate partition and **[F3-Edit]**. The following input can be made: the directory for the partition to be included in, the file system, and whether the partition is to be formatted. If the **[Format]** option is selected, all data in this partition will be deleted. The size of the partition cannot be changed later.

New partitions, or partitions created during previous installation procedures, can be deleted by selecting the partition and then selecting **[F4-Delete]**.

It is also possible to use the Logical Volume Manager during manual partitioning of hard drives. The installer will only support a single LVM media group (volume group) which is automatically named *vg_ucs*. The possibility of creating physical and logical LVM media (volumes) will only be available if LVM support was activated manually beforehand, or if a physical LVM medium is already existing on one of the connected hard drives.

When deleting LVM media (physical volumes), it should be noted that the installer will support this procedure only if all existing logical LVM media were deleted via the installer beforehand. Otherwise the contents of the physical LVM medium have to be manually moved to other physical LVM media via the *pvmove* command, and the physical LVM medium then removed from the LVM media group and deleted, via the *vgreduce* and *pvremove* commands.

The settings for a new partition are not immediately activated. The **[F5-Reset changes]** button (**F5** function key) can be used for restoring the previous settings. Only if the **[F6-Write partitions]** button (**F6** function key) is used, will the new partition table be written to the hard drive.

If there is more than one hard drive available, a dialogue is shown in which it is possible to select in which partition the bootloader should be installed.

2.10. Network configuration

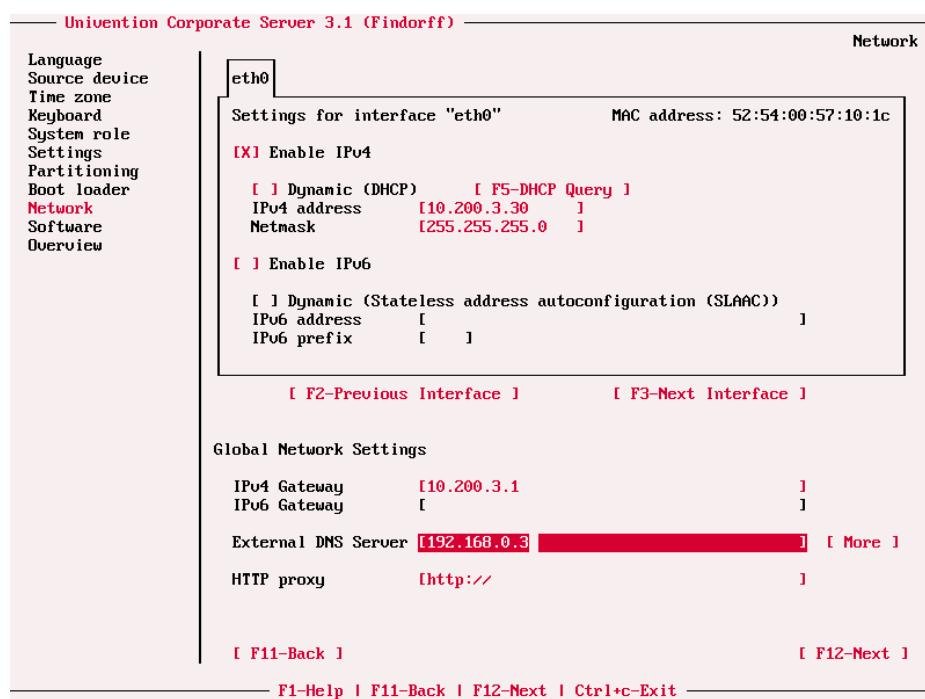
[Feedback](#)

The upper section of the network configuration displays all the network cards available in the system as a single tab (only network interfaces with the *ethX* scheme are displayed). **F2** and **F3** can be used to switch between the network cards.

If no network card is detected, the installer creates a virtual network card that can be used to continue the installation.

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., 192.168.0.10), whereas IPv6 addresses are four times as long and typically written in hexadecimal form (e.g., 2001:0DB8:FE27:0000:0000:0000:0001).

Figure 2.8. Network interface configuration



2.10.1. Configuration of IPv4 addresses

[Feedback](#)

If the **DHCP** option was not chosen, the IP address to be bound to the network card must be entered. In addition to the **IP address** the **net mask** must also be entered. **F5 DHCP request** is used to request an address from a DHCP server. Unless the **DHCP** option is activated, the values received from the DHCP request are configured statically. If the DHCP request fails, a random address from the 169.254.x.x network is used.

2.10.2. Configuration of IPv6 addresses

[Feedback](#)

The IPv6 address can be configured in two ways:

- Stateless address autoconfiguration (SLAAC) is employed in the **Dynamic** configuration, i.e. the IP address is automatically generated from the router advertisements sent from the IPv6 routers.
- 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.

Further network settings can be performed under *Global Network Settings*.

The IP addresses for the standard gateways in the subnetwork can be entered under **IPv4 Gateway** and **IPv6 Gateway**. For IPv6, a gateway must be entered in the static configuration; for the dynamic configuration, it is optional, but recommended. A gateway configured here has preference over router advertisements, which might otherwise be able to change the route.

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 only 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.

During the installation of a master domain controller only an **external DNS server** is requested, as a domain DNS server is always set up in the scope of the installation. This can then be used by the other systems in the domain. The **[More]** button offers the possibility of entering multiple name servers.

A local DNS server is also set up on the backup domain controller and slave domain controller system roles respectively. Here, you can configure which server should be primarily used for the name resolution by entering the **Domain DNS Server**. During installation, the domain DNS server of an already installed system must be specified here, e.g., that of the master domain controller.

Only a domain DNS server can be entered on member servers and base systems.

If there are no network cards apart from a virtual network card, no name server should be specified in order to avoid longer waiting times during the installation. If the name server cannot be reached during the installation, this should not be entered in the Univention installer, but instead only configured once the installation has been completed. This avoids longer timeouts during the installation.

During the installation of some of the packages, files such as the Adobe Flash plugin, have to be downloaded from external web servers due to legal restrictions. Should the UCS system have no direct access to the internet, the URL of an accessible proxy server can be entered in the **HTTP Proxy** field, which can be used for handling the download procedures. The format of the URL is `http://<IP address of the proxy server>:<Port>`.

Since, depending on the installed components, the resolution of host names during the installation procedure cannot be ensured at any time, the IP address of the proxy server should be entered.

[Feedback](#) 

2.11. Join options

This input mask does not appear with the master domain controller and base system system roles.

As standard the UCS domain is joined at the end of the installation. Deactivating the **Start join at the end of installation** option can be used to suppress the joining procedure. The system must then join the domain using the `univention-join` command.

If a UCS server system was entered in the corresponding settings mask as a name server, the name of the master domain controller is located via a DNS request. In all other cases, the **Search Domain controller Master in DNS** option must be deactivated and the fully qualified domain name of the domain controller master entered in the **Hostname of Domain controller Master** field.

A user account which is authorised to add systems to the UCS domain is called a **join account**. As standard this is the *Administrator* user, which was created during installation of the domain controller master. The corresponding password of the user account must be given under **Password**. The password must be entered identically in both fields.

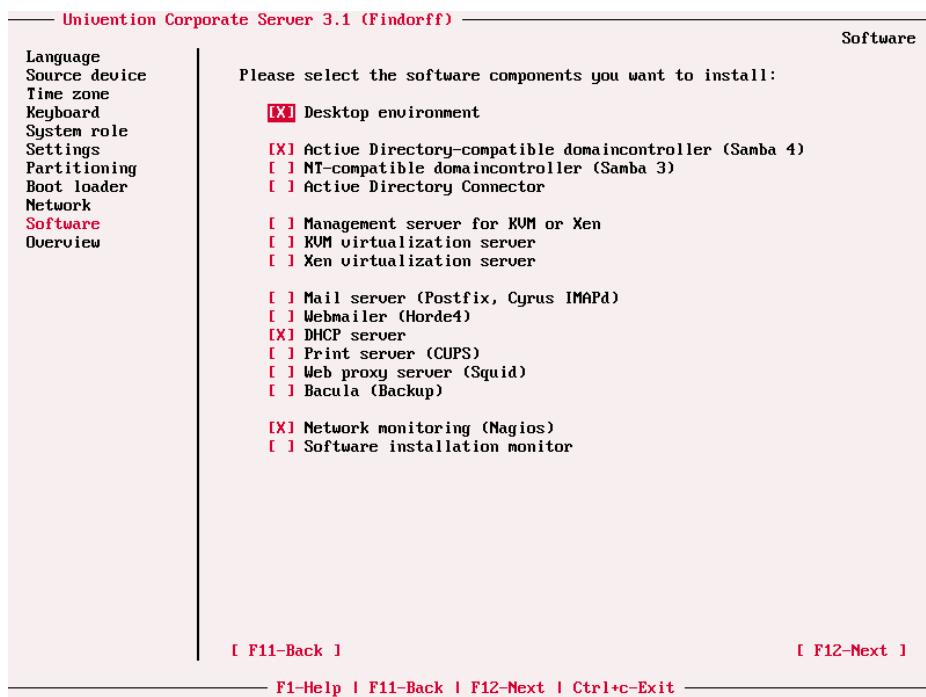
Further information on joining a domain can be found in Section 3.2.1.

2.12. Selecting software components

[Feedback](#)

The functional scope of a server system can be determined by the selection of different components. The following components are available:

Figure 2.9. Software selection



- *Desktop environment* - A desktop environment equipped for access to the web-based interfaces and comprising the X.org interface, the GDM login manager, KDE and the Firefox browser.
- *Active Directory-compatible domaincontroller (Samba 4)* - Allows the use of UCS as an Active Directory domain controller. It is not possible to install *Samba 3* and *Samba 4* on one system in parallel.
- *NT-compatible domaincontroller (Samba 3)* - Allows the use of UCS as a PDC, BDC and file server in a Windows domain based on NT domain technology
- *Active Directory Connector* - Solution for bidirectional synchronisation between the UCS directory service and Active Directory.
- *Management server for KVM or Xen* - UCS Virtual Machine Manager for the management of virtual machines
- *Xen virtualization server* - Virtualization of systems via the hypervisor Xen
- *KVM virtualization server* - Virtualization of systems using KVM

- *Mail server (Postfix, Cyrus IMAPd)* - Mail server environment based on Postfix for sending mail via SMTP and Cyrus for the provision of mail boxes via IMAP and POP3. Virus checks via the scanner ClamAV and spam detection via Spamassassin are integrated.
- *Webmailer (Horde 4)* - Web-based email access using Horde 4
- *DHCP server* - Service for the dynamic IP management for IPv4 networks.
- *Print server (CUPS)* - Print server based on the software Cups. PPD drivers for most printer models are supplied.
- *Web proxy server (Squid)* - Service for central caching and policy management of allowed websites
- *Bacula (Backup)* - Backup solution with client/server architecture for the backup and restoration of data.
- *Network monitoring (Nagios)* - System and network monitoring using the software Nagios.
- *Software installation monitor* - Central collection of installed software packages on UCS systems.

Some software components cannot be configured and are preinstalled on all system roles, e.g., the Univention Management Console, an SSH server and different command line tools and editors.

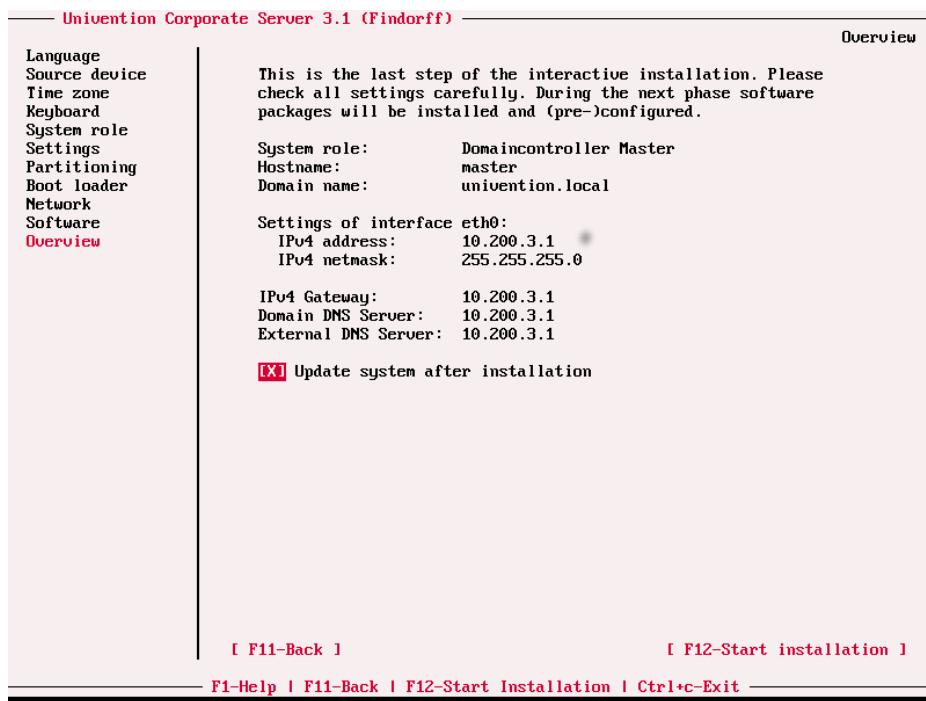
UCS is based on Debian which is the most frequently used Linux distribution in the professional sector. The advantages of Debian - and thus also of UCS - include a very flexible and well thought-out package management. The installation of further software packages is described in detail in Section 7.5.6.

[Feedback](#)

2.13. Installation overview

This dialogue shows the major settings that were made. If all the settings are correct, the installation of the software can be initiated by clicking the **[Start installation]** button, see Illustration Figure 2.10.

Figure 2.10. Installation overview



The **Update system after installation** option allows the automatic installation of available errata updates. In addition, all patch level updates available are installed on a master domain controller. On all other system roles, all the patch level updates are setup to the installation status of the master domain controller. (Univention Installer needs to login the master domain controller to check the installation status. This is done using the credentials specified in the join options).

Figure 2.11. Installation progress



During the installation, a progress bar displays the progress of the installation. The lower section of the screen provides links to further sources of information and documentation.

The system will mount the partitions to the directory tree, and install first the basic system and afterwards, if necessary, additional components and packages.

Note

During the installation of certain components, attempts will be made to download files from the internet, e.g. the Flash plugin. Should the download fail, these files can be installed at a later date. The overall installation is not impaired by this fact. However, repeated attempts to download files during installation can increase the overall installation time.

With the prerequisite that the **Start join at the end of installation** option has not been deactivated, all computers apart from base systems and domain controller masters try to join the UCS domain, read configuration settings for their services out of the LDAP directory and configure their services accordingly. Section 3.2.1 explains how the domain join can be initiated anew if the attempt to join during the installation failed.

The installation protocol of the Univention installer is stored as `installer.log.gz` and the protocol of the package installation as `installation.log.gz` in the `/var/log/univention/` directory.

For completing the installation, the Enter key is to be pressed for restarting the system. The DVD should be ejected during restart to avoid the system booting from the installation DVD again. Alternatively, the computer's BIOS can be configured in such a way that the hard drive is the first boot drive, and the CD drive the second.

The system will then boot from the hard drive. Following the boot procedure, the *root* user and *Administrator* can log on with the password specified during the installation via the command line and the *Administrator* user can do the same via the GDM log on screen.

If the computer was installed as a master domain controller, the licence can now be entered (see Section 4.2.2).

For managing the system, there is, among others, the web front-end Univention Management Console (see Chapter 4) available.

2.14. 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*.

2.15. 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].

2.16. Installation in Citrix XenServer

[Feedback](#) 

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

Chapter 3. Domain services / LDAP directory

3.1. Introduction	36
3.2. Joining domains	36
3.2.1. How UCS systems join domains	36
3.2.1.1. Subsequent domain joins with univention-join	37
3.2.1.2. Joining domains with Univention Management Console	37
3.2.1.3. Subsequent running of join scripts	37
3.2.2. How to join domains with Windows clients	37
3.2.2.1. Windows 8	38
3.2.2.2. Windows 7	39
3.2.2.3. Windows XP Professional	39
3.2.3. Ubuntu domain joins	39
3.2.4. Mac OS X domain joins	39
3.2.4.1. Domain join using the system preferences GUI	40
3.2.4.2. Domain join on the command line	40
3.3. UCS system roles	41
3.3.1. Domain controller master	41
3.3.2. Domain controller backup	41
3.3.3. Domain controller slave	41
3.3.4. member server	41
3.3.5. Base system	41
3.3.6. Ubuntu	41
3.3.7. Linux	41
3.3.8. Univention Corporate Client	41
3.3.9. Mac OS X	42
3.3.10. Managed client	42
3.3.11. Mobile client	42
3.3.12. Thin Client	42
3.3.13. Domain Trust Account	42
3.3.14. IP managed client	42
3.3.15. Windows Domaincontroller	42
3.3.16. Windows Workstation/Server	42
3.4. LDAP directory	42
3.4.1. LDAP schemas	42
3.4.1.1. LDAP schema extensions	43
3.4.1.2. LDAP schema replication	43
3.4.2. Audit proof logging of LDAP changes	43
3.4.3. Timeout for inactive LDAP connections	43
3.4.4. LDAP command line tools	43
3.4.5. Access control for the LDAP directory	44
3.4.5.1. Delegation of the privilege to reset user passwords	44
3.4.6. Name Service Switch / LDAP NSS module	45
3.4.7. Syncrep for synchronisation with non-UCS OpenLDAP servers	45
3.4.8. Configuration of the directory service when using Samba 4	45
3.5. Domain replication using the listener/notifier system	45
3.6. SSL certificate management	46
3.7. Kerberos	47
3.8. Converting a DC backup to the new DC master	48

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 authorised for him. The management system keeps the account synchronised for the windows log-in, Linux/Posix systems and Kerberos. The management of user accounts is described in Chapter 5.

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 decentralised 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:

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 2.11) or subsequently using the command `univention-join` or using the 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 synchronised 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.

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

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 authorised to add systems to the UCS domains is called a join account. By default, this is the *Administrator* user or a member of the *Domain Admins* group. 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

A domain can also be joined over the web via the **Domain join** module of the Univention Management Console.

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 the *root* user.

The username and password of a user authorised to add computers to a domain must now be entered in the resulting dialogue.

As for the domain joining procedure via the command line, a DNS service record on the DC master is also required for the Univention Management Console. There is no possibility here of entering the name of the DC master explicitly.

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

3.2.1.3. Subsequent running of join scripts

If there are join 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 on the Univention Management Console start page.

Join scripts that have not been run can be executed via the **Domain join** module in the Univention Management Console 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 scripts installed on a system. The scripts check automatically whether they have already been initiated.

The name of the join 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 as a master domain controller or backup domain controller, the user will be asked to input a username and password. This can be performed on the master domain controller and backup domain controller via the `--ask-pass` option.

3.2.2. How to join domains with Windows clients

The procedure for joining a Windows system to a UCS domain made available via Samba is now described for Windows 7/8 and Windows XP Professional. The process is similar for other Windows versions.

How to join domains with Windows clients

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.

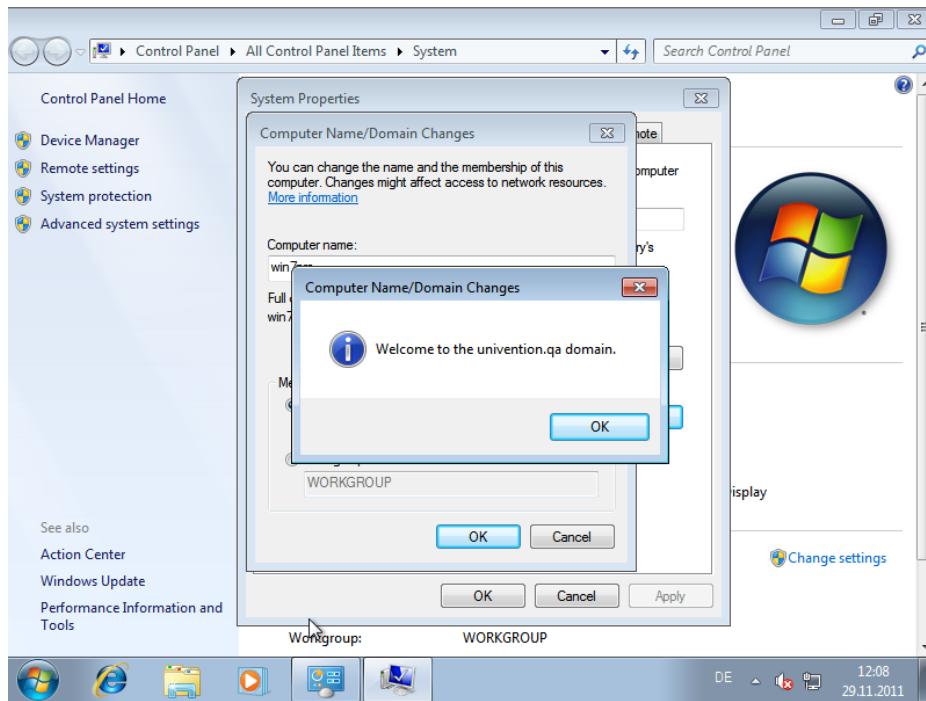
Two different versions of Samba can be used in UCS: Samba 3 implements a Windows domain based on NT domain technology, while Samba 4 implements an Active Directory directory. Further information can be found in Section 8.1.

A host account is created for the Windows client automatically when it joins the domain (see Section 7.1). 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 cancelled 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 15 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 3, it must be ensured that the NetBIOS name of the domain can be resolved. It may prove necessary to configure a WINS server for this (see Section 8.3.6).

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 synchronised with the time on the domain controller.

3.2.2.1. Windows 8

[Feedback](#)

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

When using Windows 7 to a domain based on Samba 3, certain settings must be made in the Windows registry before joining the domain. A corresponding REG file can be downloaded from <http://sdb.univention.de/1102>. The system must then be restarted. This step is not necessary with Samba 4.

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]**.

3.2.2.2. Windows 7

[Feedback](#) 

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

When using Windows 7 to a domain based on Samba 3, certain settings must be made in the Windows registry before joining the domain. A corresponding REG file can be downloaded from <http://sdb.univention.de/1102>. The system must then be restarted. This step is not necessary with Samba 4.

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]**.

3.2.2.3. Windows XP Professional

[Feedback](#) 

The dialogue for joining domains can be reached by right clicking on the **Desktop** entry in the start menu. **Properties -> Computer name -> Modify** must be selected there.

The **Domain** option field must be ticked and the name of the Samba domain entered in the input field for joining the domain. After clicking on the **[OK]** button, *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](#) 

The integration of Ubuntu clients into a UCS domain is described in Section 7.1.1.

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
```

Mac OS X domain joins

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>
```

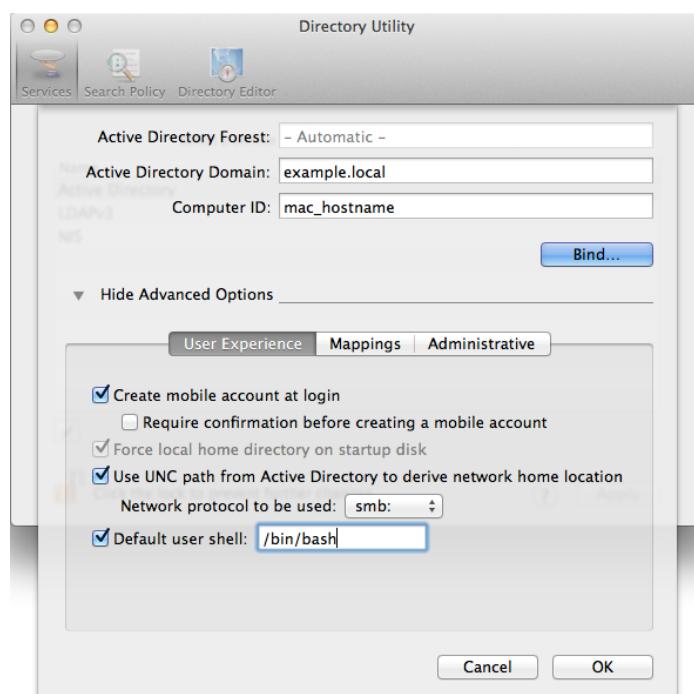
Note that the automounted shares are not displayed in the finder's sidebar.

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, 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 domainname 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 -a <mac hostname> -domain <fqdn> -ou
  "CN=Computers,<ldap_base>" \
-u <Domain Administrator> -mobile enable
```

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

3.3. UCS system roles

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

3.3.1. Domain controller master

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

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 DC. If the latter should fail, a backup DC can take over the role of the DC master permanently (see Section 3.8).

3.3.3. Domain controller slave

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 synchronised.

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

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

A base system is an autonomous system which is not a member of the domain.

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

Ubuntu clients can be managed with this system role, see Section 7.1.1.

3.3.7. Linux

This system role is used for the integration of other Linux systems than UCS and Ubuntu, e.g., for Debian or CentOS systems.

3.3.8. Univention Corporate Client

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. Managed client

[Feedback](#) 

A managed client is a stationary Linux desktop system based on Univention Corporate Desktop.

3.3.11. Mobile client

[Feedback](#) 

A mobile client is a mobile Linux desktop system based on Univention Corporate Desktop.

3.3.12. Thin Client

[Feedback](#) 

A thin client is a diskless computer which is booted over the network and displays applications run on a terminal server (Linux or Windows).

There is also no operating system installed on the thin client, which means that it requires very little maintenance.

The UCS component Univention Thin Client Services offers support for thin clients.

3.3.13. Domain Trust Account

[Feedback](#) 

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

3.3.14. 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.

3.3.15. Windows Domaincontroller

[Feedback](#) 

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

3.3.16. 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.

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

To keep the efforts required for small extensions in LDAP as low as possible, Univention Corporate Server provides its own LDAP schema 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).

3.4.1.2. LDAP schema replication

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

The `univention-directory-logger` package allows logging of all changes in the LDAP directory service. In addition, an integrated hash sum ensures that no changes can be deleted from the log file. This is logged in the `/var/log/univention/directory-logger.log` file.

3.4.3. Timeout for inactive LDAP connections

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 since UCS 3.0.

3.4.4. LDAP command line tools

In addition to the Univention Management Console 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

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.

The default setting of the LDAP server after new installations with UCS does not allow anonymous access to the LDAP directory. This behaviour 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 is 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

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/aceslist/groups/dn`. The LDAP server must be restarted after making changes.

Passwords can be reset via Univention Directory Manager. In the default setting, Univention Directory Manager 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 can be linked to the members of the *User Password Admins* group and/or a corresponding container in the LDAP directory.

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.

To prevent the resetting of the passwords for certain users (e.g., domain administrators), the UIDs of the users to be protected can be entered, separated by commas, in the Univention Configuration Registry variable

`ldap/acl/user/passwordreset/protected/uid`. Once a variable has been changed, it is necessary to restart the LDAP directory service using the `/etc/init.d/slappd restart` command for the changed LDAP ACLs to take effect. In the default setting, the *Administrator* user is protected against having his password changed by the *User Password Admins* group.

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 synchronisation with non-UCS OpenLDAP servers

[Feedback](#) 

The Syncrepl replication service can also be activated parallel to the notifier service for the synchronisation 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 as of UCS 3.0 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.

Managed/mobile clients or member servers using UCS 2.4-4 continue to use the ports 389/636. Such systems must either use a UCS 2.4-4 domain controller or a UCS 3.x domain controller without Samba 4 as their LDAP server.

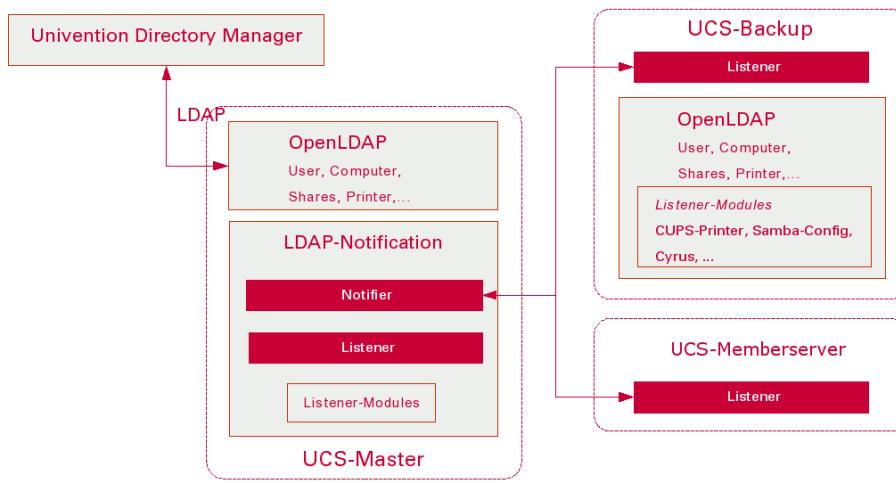
3.5. Domain replication using the listener/notifier system

[Feedback](#) 

Replication of the directory data within a UCS domain occurs via the Univention Directory Listener/Notifier mechanism: On the master domain controller the notifier service monitors changes in the LDAP directory and makes the selected changes available transaction-based to the listener services on the other domain systems

with a copy of the LDAP. Alongside replication of the LDAP contents, the dissemination of the domain-wide file changes also includes transfer of the changes to LDAP contents in configuration files of non-LDAP-compatible services (e.g., NFS).

Figure 3.3. Listener/Notifier mechanism



If, for example, a new NFS share is created in the domain management of the Univention Management Console, a Python script - a so-called Univention Directory listener module - is used to integrate another line in the `/etc/exports` file. The file is also adapted if the share is removed or modified.

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 the 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 **Basic settings** on the **Certificate** tab.

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-sslchange].

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 synchronised with the CA on the domain controller master by a Cron job every 20 minutes.

Caution

The CA is synchronised 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 3.8), 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 the 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).

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` (host certificate) and `ssl/validity/root` (root certificate). The values entered there reflect the number of days since the 1.1.1970.

On the Univention Management Console overview page - which can be accessed by clicking on the question mark in the top toolbar - the effective expiry date of the computer and root certificate is also displayed under **About UMC**.

3.7. Kerberos

[Feedback](#) 

Kerberos is an authentication framework the purpose of which is to permit secure identification in the potentially insecure connections of decentralised networks. In Kerberos, all clients use a foundation of mutual trust, the *Key Distribution Centre* (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 can be defined via the Univention Configuration Registry variable `kerberos/realm`.

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

Univention Corporate Server uses the Heimdal Kerberos implementation. Several domain controllers can be operated as a KDC within one domain; by default, each domain controller works as a KDC. The KDC used by a system can be reconfigured via the Univention Configuration Registry variable `kerberos/kdc`. As standard, the KDC is selected via a DNS service record.

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. Converting a DC backup to the new DC master

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, no writing changes can be performed.

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

- In an emergency if the hardware of the master domain controller fails
- When the plan is to replace the master domain controller with new hardware

Caution

The conversion cannot be reversed. If the previous master is to be used again, it must be reinstalled and joined as a backup domain controller for it to be used again. The promotion of a backup domain controller to a master domain controller is a far-reaching configuration step and should be prepared thoroughly!

The conversion primarily involves the changeover of the services relevant for authentication such as LDAP, Kerberos and Samba. The installed software needs to be adjusted manually (this can be done using the **Basic settings** module of the Univention Management Console). For example, if the mail component was installed on the previous DC master, it is not automatically on the new DC master after the conversion.

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 subsequent comparison of the installed packages. The package list can be created with the following command:

```
COLUMNS=200 dpkg --list > packagelist.txt
```

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. This can be achieved with the following command:

```
ucr dump > ucr.txt
```

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

The computer name and/or the IP address of the master domain controller is automatically changed to the new name in all configurations managed from the UCS LDAP or via Univention Configuration Registry in the scope of the promotion. If the name is referenced in configuration files or on systems which are not managed by Univention Configuration Registry, it must be adapted accordingly following the promotion.

Chapter 4. Central domain management with UMC

4.1. Introduction	49
4.2. Operating instructions for Univention Management Console	50
4.2.1. Login	50
4.2.2. Activation of UCS license / license overview	51
4.2.3. Operating instructions for domain modules	51
4.2.3.1. Searching for objects	52
4.2.3.2. Creating objects	53
4.2.3.3. Editing objects	53
4.2.3.4. Deleting objects	53
4.2.3.5. Moving objects	53
4.2.4. Favorites	53
4.2.5. Feedback on UMC and UCS	54
4.3. Collection of usage statistics	54
4.4. LDAP directory browser	54
4.5. Policies	55
4.5.1. Creating a policy	55
4.5.2. Applying policies	56
4.5.3. Editing a policy	56
4.6. Expansion of the attributes managed in UMC with extended attributes	56
4.7. Structuring of the domain with user-defined LDAP structures	60
4.8. Command line interface of domain management (Univention Directory Manager)	61
4.9. Evaluation of data from the LDAP directory with Univention Directory Reports	63
4.9.1. Creating reports in the Univention Management Console	64
4.9.2. Creating reports on the command line	64
4.9.3. Adjustment/expansion of Univention Directory Reports	64

4.1. Introduction

[Feedback](#) 

Univention Management Console (UMC) is the central tool for web-based administration of the UCS domain. UMC supports two basic module types:

- *Domain modules* manage all the data saved in the LDAP directory, e.g., users, groups or host accounts. All the modifications made here apply to the whole domain. Domain modules are only displayed on the master domain controller and backup domain controller. All the domain modules can also be controlled with a command line interface, the Univention Directory Manager.
- *System modules* allow the configuration or management of an individual system, e.g., the starting or stopping of services or the configuration of Univention Configuration Registry variables.

Section 4.2 describes the user interface of the Univention Management Console. Section 4.6 describes how the functions of the domain management can be expanded with additional attributes.

The domain data can also be exported and evaluated with Univention Directory reports, see Section 4.9.

The Univention Directory Manager - the command line interface of the domain management is described in the Section 4.8.

4.2. Operating instructions for Univention Management Console

4.2.1. Login

Univention Management Console uses numerous Javascript and CSS functions to display the web interface. Cookies need to be permitted in the browser. The following browsers are supported.

- Chrome as of version 14
- Firefox as of version 3.6 (version 10 or above is recommended due to performance reasons)
- Internet Explorer as of version 8 (version 9 or above is recommended due to performance reasons)
- Safari (on the iPad 2, iOS version 6.1.2)
- Android browser (version 4.1 or above)

Users with older browsers may experience display problems.

Univention Management Console is available in German and English; the language to be used can be selected during login.

The web interface of the Univention Management Console can be opened via the URL `https://server-name/umc/`. Alternatively, access is also possible via the server's IP address.

Under certain circumstances it may be necessary to access the Univention Management Console 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!

Figure 4.1. UMC login mask



Welcome to Univention Management
Console. Please enter your domain
username and password for login.

Username
Administrator

Password

Language
English

Login

Once the URL has been opened, a login mask is displayed in which the **Username** and **Password** need to be entered:

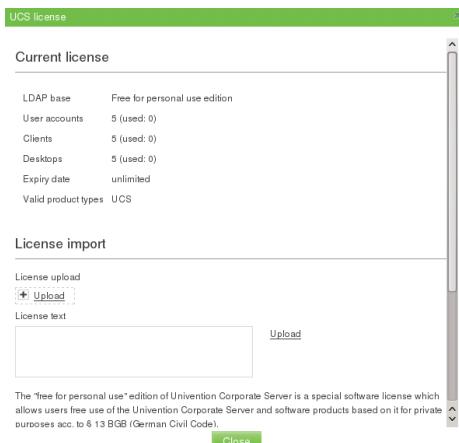
- When logging in with the system's local *root* account (see Section 7.4.1) only the system modules of the Univention Management Console are displayed.
- When logging in with the *Administrator* account on the master domain controller or backup domain controller all domain modules are displayed.
- When logging on with another user account, the domain modules approved for the user are shown.

Following ten minutes of inactivity, the browser session is automatically closed and it is necessary to login again. This interval can be adjusted in seconds with the Univention Configuration Registry variable `umc/http/session/timeout`.

4.2.2. Activation of UCS license / license overview

The UCS license can be set up by clicking on the gear-wheel symbol in the top right line of the screen. To do so, the **License** menu entry must be selected. The current license status is shown under **Current license**.

Figure 4.2. Displaying the UCS license

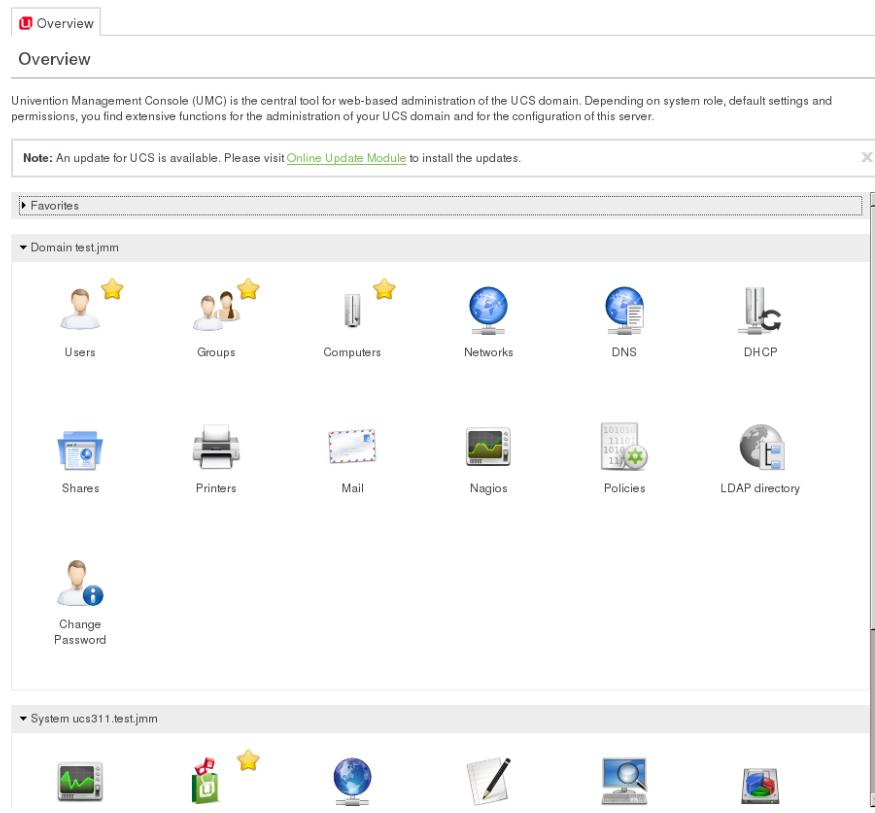


License upload can be used to select a license file. Alternatively, the license key can be copied into the **License text** input field and activated with **Upload**.

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

4.2.3. Operating instructions for domain modules

All of the domain modules in Univention Management Console are controlled identically from a structural perspective. The following examples are presented using the user management but apply equally for all modules.

Figure 4.3. Module overview


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

- Users - Chapter 5
- Groups - Chapter 6
- Computers - Chapter 7
- Networks - Section 9.1
- DNS - Section 9.2
- DHCP - Section 9.3
- Shares - Chapter 10
- Printers - Chapter 11
- E-mail - Chapter 12
- Nagios - Chapter 13

The use of policies (Section 4.5) and the LDAP navigation (Section 4.4) are described separately.

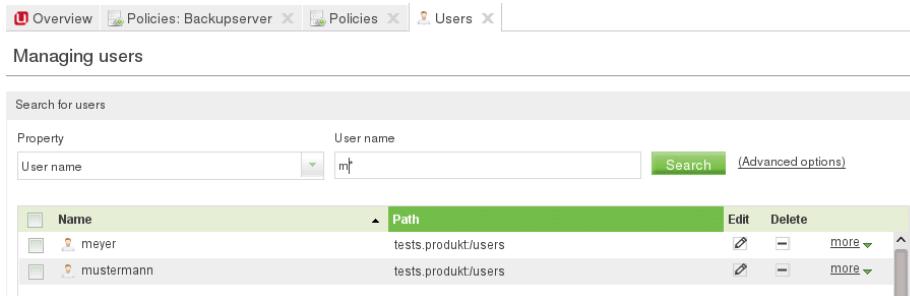
4.2.3.1. Searching for objects

[Feedback](#) 

The module overview lists all the objects managed by this module. The **Property** selection field can be used to select an attribute. The objects can then be filtered according to this attribute. The search filter is entered in

the next input field. For example, in user management, the selection of *Username* and the search maske *m** can be used to display only the user names beginning with *m*.

Figure 4.4. Searching for users



The screenshot shows the 'Users' module in the UMC. At the top, there are tabs for 'Overview', 'Policies: Backupserver', 'Policies', and 'Users'. Below the tabs, a search bar is labeled 'Search for users' with a dropdown menu set to 'User name'. A search input field contains 'm*' and a 'Search' button. To the right of the search bar is a link '(Advanced options)'. The main area is titled 'Managing users' and contains a table with two rows of user data. The columns are 'Name', 'Path', 'Edit', and 'Delete'. The first row has a selection checkbox, the name 'meyer', the path 'tests.produkt/users', and edit/delete buttons. The second row has a selection checkbox, the name 'mustermann', the path 'tests.produkt/users', and edit/delete buttons.

Clicking on **Advanced options** can be used to select another **Search in** selection field. This selection can be used to choose whether the complete LDAP directory or only selected LDAP containers are searched in. Further information on the structure of the LDAP directory service can be found in Section 4.7.

4.2.3.2. Creating objects

[Feedback](#)

The lowest line of the screen includes an action line for creating a new object. The menu entries have the same names as the module name, e.g., **Add user**. If a user template is defined, it can be selected (see Section 5.6).

4.2.3.3. Editing objects

[Feedback](#)

Clicking on the stylised pen behind an LDAP object in the search overview allows to edit the object. The individual attributes are described in the individual documentation chapters. Clicking on **Save changes** accepts all the changes made in the LDAP directory; **Back to search** cancels the editing.

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 stylised 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.

4.2.3.4. Deleting objects

[Feedback](#)

Clicking on the stylised minus sign behind an LDAP object in the search overview 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.

4.2.3.5. Moving objects

[Feedback](#)

The **more** selection field behind an LDAP object in the search overview can be used, after the selection of **Move 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.

4.2.4. Favorites

[Feedback](#)

Commonly used UMC modules are shown in a submenu **Favorites**. If you move the mouse right over the icon of a module already marked as a favorite, a translucent red cross appears. If you click this, the module

is removed from your favorites. To add a module to the favorites, the yellow star above the modules should be activated in the same way.

4.2.5. Feedback on UMC and UCS

[Feedback](#) 

The **Feedback** option can be reached by clicking on the question mark symbol in the top menu bar. Here you can provide feedback on UCS and the Univention Management Console via a web form.

4.3. Collection of usage statistics

[Feedback](#) 

Anonymous usage statistics on the use of the Univention Management Console are collected when using the *free for personal use* version of UCS (which is generally used for evaluating UCS). The modules opened are logged in an instance of the web traffic analysis tool Piwik. This makes it possible for Univention to tailor the development of the Univention Management Console better to customer needs and carry out usability improvements.

This logging is only performed when the *free-for-personal-use license* is used. The license status can be verified by clicking on the cog symbol in the top righthand corner of the Univention Management Console and selecting **License**. If **Free for personal use edition** is listed under **LDAP base**, this version is in use. When a regular UCS license is used, no usage statistics are collected.

Independent of the license used, the statistics generation can be deactivated by setting the Univention Configuration Registry variable `umc/web/piwik` to *false*.

4.4. LDAP directory browser

[Feedback](#) 

The **LDAP directory** domain 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 4.5. Navigating the LDAP directory

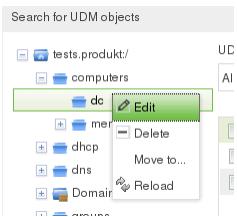
Name	Type	Path	Edit	Delete
slave	Domain Controller Slave	jmm.test/computers/dc		
ucs311	Domain Controller Master	jmm.test/computers/dc		

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 LDAP object** option can be used to add new objects here too. Similar to the control elements described in Section 4.2, existing objects can also be edited, deleted or moved here.

Figure 4.6. Editing LDAP container settings



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

4.5. 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.

4.5.1. Creating a policy

[Feedback](#) 

Policies can be managed via the **Policies** module of the Univention Management Console. The operation is the same as for the functions described in Section 4.2.

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.

4.5.2. 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 4.4).
- A **Policies** tab is shown in the individual domain modules for which there are policies, e.g., for users. A different policy can be specified here for this user.

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.

4.5.3. Editing a policy

[Feedback](#) 

Policies can be edited and deleted in the UMC domain module **Policies**. The interface is described in Section 4.2.

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.

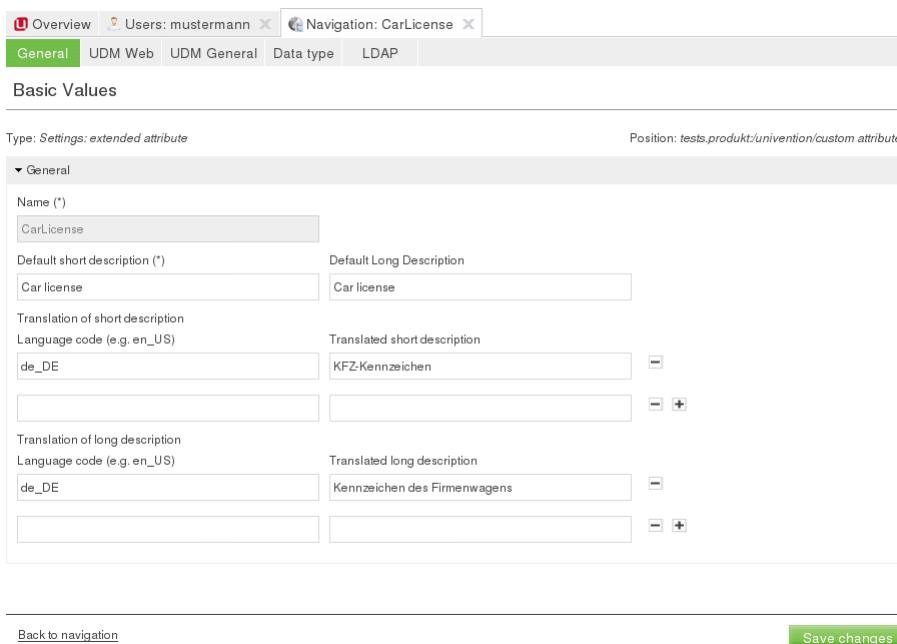
The policy tab of the individual LDAP objects also includes the **edit** option, which can be used to edit the policy currently applicable for this object.

4.6. Expansion of the attributes managed in UMC with extended attributes

[Feedback](#) 

The domain management of the 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 **LDAP directory** module of the Univention Management Console. 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 LDAP object**.

Figure 4.7. Extended attribute for managing a car license


Type: Settings: extended attribute Position: tests produkt/univention/custom attributes

General

Name (*)

CarLicense

Default short description (*) Default Long Description

Car license Car license

Translation of short description

Language code (e.g. en_US)

de_DE KFZ-Kennzeichen

Translated short description

Translation of long description

Language code (e.g. en_US)

de_DE Kennzeichen des Firmenwagens

Translated long description

Save changes

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

Attribut	Beschreibung
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.
Default short description	Used as title of the input field in the Univention Management Console or as the attribute description in the command line interface.
Default long description	This long description is shown as a tool tip in the input fields in Univention Management Console.
Translated 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., <i>de_DE</i> or <i>fr_FR</i>) in this input field.
Translated 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 assigning the respective long description to a language code (e.g., <i>de_DE</i> or <i>fr_FR</i>) in this input field.

Table 4.2. UDM Web

Attribut	Beschreibung
Tab name	The name of the tab in the Univention Management Console on which the extended attribute should be displayed. New tabs can also be added here.

Expansion of the attributes managed in UMC with extended attributes

Attribut	Beschreibung
	If no tab name is entered, <i>user-defined</i> will be used.
Position number on tab	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 in question in ascending order of their numbers. If extended attributes have the same number, they are listed in a random order. If the difference between two consecutive numbers is larger than 1 and the first attribute is shown in the left column, the second will be displayed in the next line. If the <i>Overwrite existing tab</i> option is activated, the numbers will be used to overwrite an existing input field. The numbering of the input fields starts in the top left of the tab with 1.
Translated tab name	Translated tab names can be assigned to the corresponding language code (z.B. <i>de_DE</i> oder <i>fr_FR</i>) in this input field.
Group name	Groups allow the structuring of a tab. A group is separated by a grey horizontal bar and can be shown and hidden. If no group name is specified for an extended attribute, the attribute is placed above the first group entry.
Position number of group	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.
Translated 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., <i>de_DE</i> or <i>fr_FR</i>).
Overwrite existing widget	In some cases it is practical to overwrite predefined input fields with extended attributes. If this option is activated, the input field located at the specified number will be overwritten with this extended attribute. It must be noted that this option can cause problems with compulsory fields.
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.
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.
Advanced tab	Settings possibilities which are not often used can be placed in the extended settings tab.
Do not show in UDM Web	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.

Table 4.3. UDM General

Attribut	Beschreibung
UDM CLI name of extended attribute	The specified attribute name should be used when employing the command line interface. When the extended attribute is saved, the <i>Name</i> of the <i>General</i> tab is automatically adopted and can be subsequently modified.
Hook	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 at http://sdb.univention.de/1080 (currently only available in German).
Options	Some extended attributes can only be used practically if certain object classes are activated on the (<i>Options</i>) tab. One or more options can optionally be saved in this input field so that this extended attribute is displayed or editable.
Needed module	The Univention Directory Manager module which is to be expanded with the extended attribute. An extended attribute can apply for multiple modules.

Table 4.4. Data type

Attribut	Beschreibung
Syntax	<p>When values are entered, the Univention Directory Manager performs a syntax check.</p> <p>Apart from standard syntax definitions (<i>string</i>) and (<i>integer</i>), there are three possibilities for expressing a binary condition. The syntax <i>True-False</i> is represented at LDAP level using the strings <i>true</i> and <i>false</i>, the syntax <i>TrueFalseUpper</i> corresponds to the OpenLDAP boolean values <i>TRUE</i> and <i>FALSE</i> and the syntax <i>boolean</i> does not save any value or the string <i>I</i>.</p> <p>The syntax <i>string</i> is the default.</p>
Default value	If a preset value is defined here, new objects to be created will be initialised 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 schema definition of the LDAP attribute specifies whether one or several instances of the attribute may be used in one LDAP object.
May change	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 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.
Unsearchable	<p>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.</p> <p>This is only needed in exceptional cases.</p>

Attribut	Beschreibung
Value not editable	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.

Table 4.5. LDAP

Attribut	Beschreibung
Object class	<p>Object class to which the attribute entered under <i>LDAP mapping</i> belongs.</p> <p>Predefined LDAP schema extensions for extended attributes are provided with the object class <i>univentionFreeAttributes</i>. Further information can be found in Section 3.4.1.1.</p> <p>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.</p>
LDAP mapping	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.
Delete object class	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 <i>Objektklasse</i> will also be removed from the LDAP object if this option is activated.

4.7. Structuring of the domain with user-defined LDAP structures

[Feedback](#)

Containers and organisational 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:

- Organisational 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 organisational units are managed in the **LDAP directory** module of the Univention Management Console and are created with **Add LDAP object** 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

Attribut	Beschreibung
Name	A random name for the container / organisational unit.
Description	A random description for the container / organisational unit.

Table 4.7. 'Advanced settings' tab

Attribut	Beschreibung
Add to standard <object type> containers	If this option is activated, the container or organisational 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

Attribut	Beschreibung
	The tab is described in Section 4.5.2.

[Feedback](#)

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

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 Console 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.

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
[...]
```

Command line interface of domain management (Univention Directory Manager)

```

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 `posix` and `person` are 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:

```
univention-directory-manager users/user modify \  
  --option="posix" --option="mail" --option="kerberos"
```

- `--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/modulename> <Operation> \  
  --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"
```

4.9. 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 the 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 the Univention Management Console web interface.

4.9.1. Creating reports in the Univention Management Console

[Feedback](#) 

To create a report, you need to switch to the domain module for users, groups or hosts. The report is created for all the objects shown in the overview. The selection can be limited based on a **Property** using filters. Clicking on **Create report** allows to choose between the **Standard Report** in PDF format and the **Standard CSV Report** in CSV format.

Figure 4.8. 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.

4.9.2. 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
```

4.9.3. 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. User management

5.1. User management in the Univention Management Console	65
5.2. User password management	72
5.3. Password settings for Windows clients when using Samba	73
5.4. Password changes by users via Univention Management Console	74
5.5. Automatic lockout of users after failed login attempts	74
5.6. User templates	75

UCS integrates central identity management according to the *same user, same password* principle. All user information are managed centrally in UCS via the Univention Management Console and stored in the LDAP directory service.

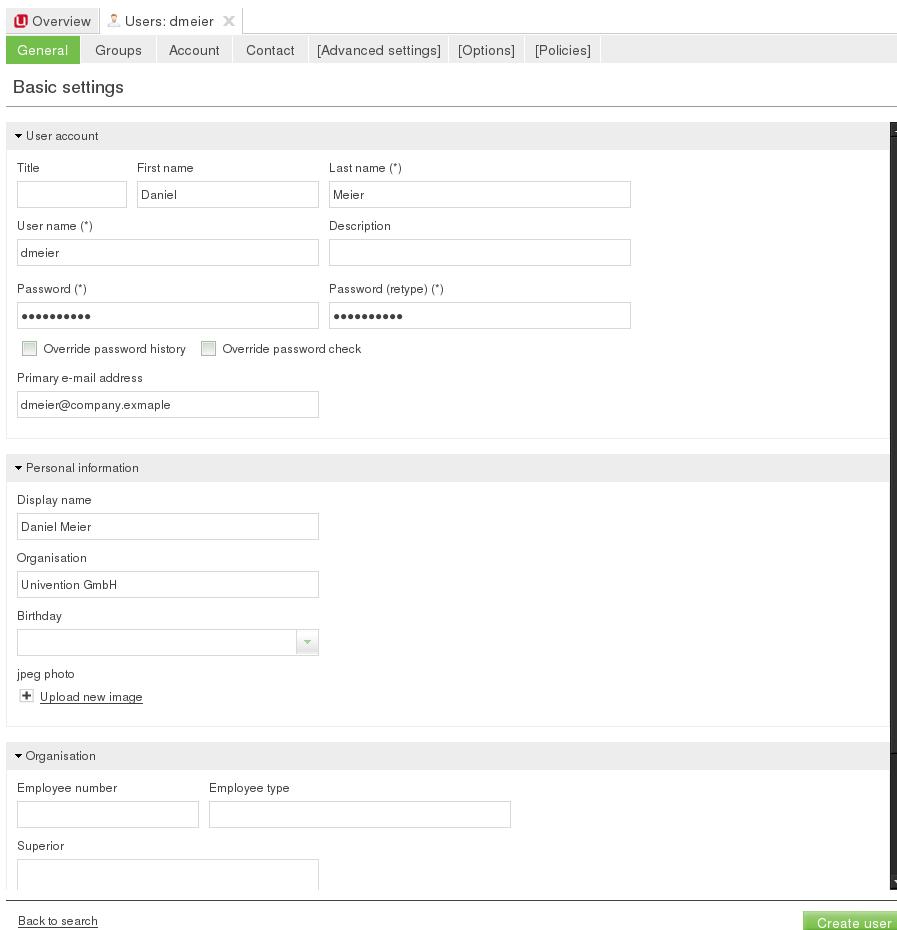
All the services integrated in the domain access the central account information, i.e., the same username and password are used for the user login to a Windows client as for the login on the IMAP server.

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

5.1. User management in the Univention Management Console

[Feedback](#) 

Users are managed in the *Users* module of the Univention Management Console.

Figure 5.1. Creating a user in UMC


The screenshot shows the 'General' tab of the Univention Management Console (UMC) for creating a new user. The user is named 'dmeier'. The 'Basic settings' section contains fields for Title (empty), First name (Daniel), Last name (Meier), User name (*), Description (empty), Password (*), Password (retype) (*), Primary e-mail address (dmeier@company.example), and checkboxes for Override password history and Override password check. The 'Personal information' section includes fields for Display name (Daniel Meier), Organisation (Univention GmbH), Birthday (empty), and jpeg photo (with a link to upload). The 'Organisation' section includes fields for Employee number (empty) and Employee type (empty). The 'Create user' button is located at the bottom right.

General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

Table 5.1. 'General' tab

Attribute	Description
User name	<p>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 underscores.</p> <p>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 <i>smith</i> already exists, then the user name <i>Smith</i> cannot be created.</p> <p>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 <code>directory/manager/user_group/uniqueness</code> is set to <i>false</i>, this check is removed.</p>
Description	Arbitrary descriptions for the user can be entered here.
Password	The user's password has to be entered here.

Attribute	Description
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 5.2.
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 5.2.
Primary e-mail address	The e-mail address of the user is declared here, see Section 12.3.2.
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 synchronisation with Active Directory and Samba 4 among other things.
Organisation	The organisation is to be entered here.
Birthday	This field is used to save a user's birthday.
jpeg photo	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 5.2. '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 <i>Domain Users</i> is preset.
Groups	Here it is possible to set further group memberships for the user in addition to the primary group.

Table 5.3. 'Account' tab

Attribute	Description
Account deactivation	The Account deactivation selection field can be used to deactivate the user account for one or more login methods. As long as the respective account type is deactivated, the user cannot log into the system. This is typically used when a user leaves the company. In a heterogeneous environment, an account deactivation might also be caused by external tools; in that case the selection field reflects the account status. Normally users should always be deactivated for all account types. The following deactivation states can be realised:

Attribute	Description
	<ul style="list-style-type: none"> • None - Basic status; all logins are possible. • All disabled - All account types are blocked. • Windows disabled • Kerberos disabled • POSIX disabled • Windows und POSIX disabled • Windows und Kerberos disabled • POSIX und Kerberos disabled <p>The following interconnections between the different login methods are derived from the UCS PAM configuration:</p> <ul style="list-style-type: none"> • The Linux login (e.g., on GDM or a tty) is only deactivated if all login methods are deactivated; a deactivated POSIX account alone is not enough. • Samba requires a not-deactivated POSIX account - that means that the deactivation of the POSIX account automatically deactivates Samba as well. • The Kerberos library (Heimdal) also evaluates the Samba account settings - that means that the deactivation of the Windows account will also deactivate Kerberos.
Locked login methods	<p>This selection field can be used to block individual login methods. This can happen automatically for security reasons, for example, if a user has entered his password incorrectly too often, see Section 5.5.</p> <p>Normally users should always be blocked for all login methods.</p> <p>In contrast to Account deactivation, the account is not blocked, but the login is denied. The following login methods can be restricted:</p> <ul style="list-style-type: none"> • None • Locked all login methods • Locked Windows/Kerberos only • Locked POSIX/LDAP only
Account expiry date	<p>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.</p> <p>If the date is deleted or replaced by a different, future date, the user will regain the right to log in.</p>
Password expiry date	<p>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 5.2.</p>

Attribute	Description
	If a password expiry interval is defined, the password expiry date is automatically adjusted when passwords are changed. If no Expiry interval is declared, the old expiry date will be deleted and no new date will be set.
Change password on next login	If this checkbox is ticked, then the user has to change his password during the next login procedure.
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. <i>M:</i> .
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. <code>\ucs-file-server\smith</code>
Windows logon script	The user-specific logon script relative to the Netlogon share is entered here, e.g. <code>scripts\user.bat</code> .
Windows profile directory	The profile directory for the user can be entered here, e.g. <code>\ucs-file-server\user\profile</code> .
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.
Samba User Login Times	This input field contains time periods for which this user can log on to Windows computers. 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, <code>/bin/bash</code> 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. 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 grey and barred from change.
Group ID	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.

Attribute	Description
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 5.4. 'Contact' tab

Attribute	Description
E-Mail address(es)	Additional e-mail addresses can be saved here. These are not evaluated by the mail server. The values of this attribute are stored in the LDAP attribute <i>mail</i> . 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.
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	The private fixed network phone number can be entered here.
Mobile telephone number	The user's mobile numbers can be entered here.
Pager telephone number	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 5.5. 'Mail' tab

This tab is displayed in the advanced settings.

The settings are described in Section 12.3.2.

The following tab configures settings for working on Windows terminal servers.

Table 5.6. 'Windows Advanced' tab (advanced settings)

Attribute	Description
Home directory for Windows terminal services	The directory path which is to be the user's Windows home directory on the terminal server can be entered here, e.g. <code>\ucs-file-server\ts\user</code> .
Home drive for Windows terminal services	If the Windows home directory for this user is to appear on a different Windows drive than specified in the Samba configuration, the respective letter of the drive can be entered here followed by a colon, e.g. <code>M:</code> .
Startup command	Path to a program which should be run when a terminal session is started.
Working directory for startup command	The program's working directory, which is entered under Startup command .
Use client configuration for startup command	Both configuration settings Startup command and Working directory for startup command can be overwritten by the client application. If this checkbox is activated, the client configuration is used.

Attribute	Description
Connect client drives at login	The drives of the connecting client computer can be made available during a terminal server session when this checkbox is activated.
Connect client printers at login	The client printers are connected during log-in to the terminal server and are thus available during the terminal server session.
Make client default printer the default printer for Windows terminal services	If this checkbox is activated the client standard printer will be declared the standard printer for this terminal server session.
Allow Windows terminal server login	If this checkbox is activated the user can log on to a terminal server.
CTX Mirroring	This selection list specifies whether a user session can be mirrored. If Disabled is selected the session cannot be mirrored.
Terminated or timed-out sessions	In this selection list one can select whether ended or expired connections should be Disconnected ed or Reset .
Reconnect session	Here you can select whether the ended connection to each client or just the previous client can be rebuilt.
CTX RAS Dialin	This option configures the callback function of a remote access server. If enabled, the dialin line of the user is disconnected after authentication and the user is called back. If an entry with the Input On option is selected, the user who initiated the mirroring will be given the permission to perform keyboard inputs and mouse action in the mirrored session. If an entry with the Message On option is selected, a message is shown on the client stating that a request has been made to mirror the session.
Profile directory for Windows terminal services	The path to the Windows profile which is to be used in the terminal server session should be entered here. If no value is entered, the standard profile path is used.
Keyboard layout	The keyboard layout for the terminal server session.

Table 5.7. '(Options)' tab

Attribute	Description
Mail account	If this checkbox is not ticked, the user will not be assigned the object class <i>univentionMail</i> .
Kerberos principal	If this checkbox is not ticked, the user will not be assigned the object classes <i>krb5Principal</i> and <i>krb5KDCEntry</i> .
Samba account	If this checkbox is not ticked, the user will not be assigned the object class <i>sambaSamAccount</i> .
POSIX account	If this checkbox is not ticked, the user will not be assigned the object classes <i>posixAccount</i> and <i>shadowAccount</i> .
Personal information	If this checkbox is not ticked, the user will not be assigned the object classes <i>organizationalPerson</i> and <i>inetOrgPerson</i> .
Public key infrastructure account	If this checkbox is not ticked, the user will not be assigned the object class <i>pkiUser</i> .
Simple authentication account	This option can be used for creating user objects which have only a username and a password. With these users, authentication is possible against the LDAP directory service exclusively; logging into UCS or

Attribute	Description
	Windows systems is not possible. If this option is activated, the object classes <i>uidObject</i> and <i>simpleSecurityObject</i> will be used.

5.2. User password management

[Feedback](#)

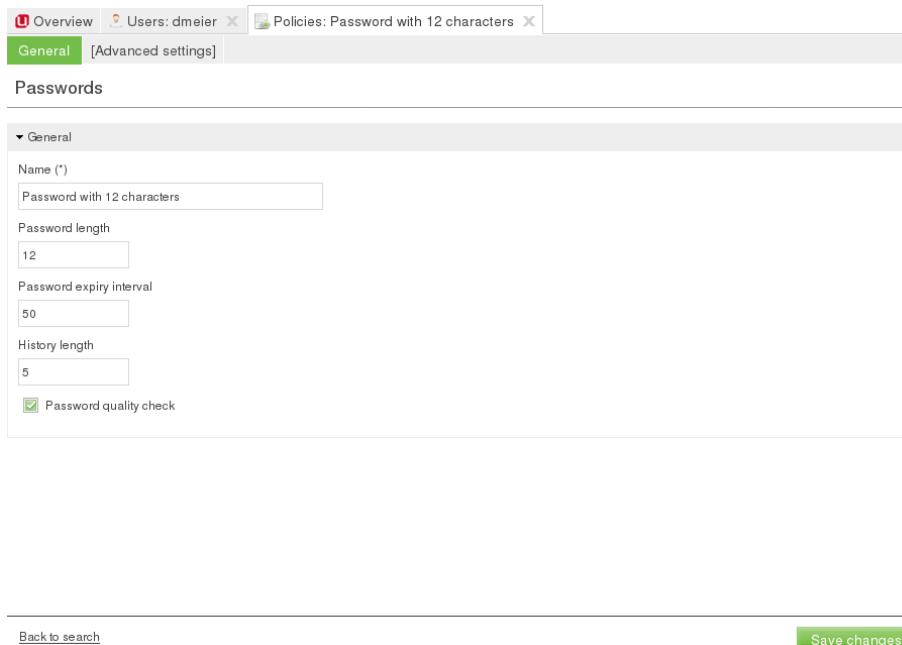
Passwords which are difficult to guess and regular password changes are an essential element of the system security of a UCS domain. The following properties can be configured for users using a *password* policy. If Samba is used, the settings of the Samba domain object (see Section 5.3) apply for logins 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 in the UCS PAM configuration.

Figure 5.2. Configuring a password policy



Overview Users: dmeier Policies: Password with 12 characters

General [Advanced settings]

Passwords

General

Name (*)
Password with 12 characters

Password length
12

Password expiry interval
50

History length
5

Password quality check

Back to search Save changes

- 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.

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.
- A *password expiry interval* demands regular password changes. A password change is demanded during logons to the 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`).

5.3. 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 *LDAP directory* module of the Univention Management Console. 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 5.2) should be set identically, otherwise different password requirements will apply for logins to Windows and UCS systems.

Table 5.8. 'General' tab

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

Attribute	Description
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.

5.4. Password changes by users via Univention Management Console

[Feedback](#) 

As standard, users cannot change their password via the Univention Management Console. To allow user-initiated password changes, the modification must be allowed in the LDAP ACLs and in the Univention Management Console.

To adapt the LDAP ACLs, the Univention Configuration Registry variable `ldap/acl/user/password/change` must be set to `yes` on the master domain controller and the LDAP server restarted.

For simple activation in the UMC, the policy `default-udm-self` can be linked with an LDAP object (e.g., with the LDAP base or the group *Domain Users* (see Section 4.5.2)).

5.5. Automatic lockout of users after failed login attempts

[Feedback](#) 

As standard, a user can enter his 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. The lockout is activated locally per computer system as standard. In other words, if a user enters his password incorrectly too many times on one system, he 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. The global lock can only be set on domain controller master/backup systems as other system roles do not possess the requisite permissions in the LDAP directory. On these system roles, the user is, however, locally locked or unlocked again via the listener module.

Caution

This setting can also be misused, for example when a user has locked his screen and another user enters the password incorrectly several times in his absence. In this case, the user must contact the administrator to have his account unlocked.

The automatic lockout of users following failed logins can be activated 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 activated is configured in the Univention Configuration Registry variable `auth/faillog/limit`. The counter is reset each time the password is entered correctly.

As standard, 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`.

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 interval has elapsed. This is done by specifying a time period in seconds in Univention Configuration Registry variable `auth/faillog/unlock_time`. If the value is set to 0, the lock is reset immediately.

If accounts are locked locally, the administrator can unlock a user by entering the command `faillog -r USERNAME`. If the lock occurs globally in the LDAP, the user can be reset in the **User account** tab of a user in Univention Directory Manager.

5.6. User templates

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 5.3. Selecting a user template



User templates are administrated in the **LDAP directory** module of the Univention Management Console. 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 LDAP object** 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>@firma.com. Substitutions are generally possible 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 brackets 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>@firma.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 brackets. 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.

User templates

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.

It is also possible to combine options, e.g: `<:umlauts,upper>`.

Chapter 6. Group management

6.1. Managing groups in the Univention Management Console	77
6.2. Nested groups	79
6.3. Local group cache	79

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:

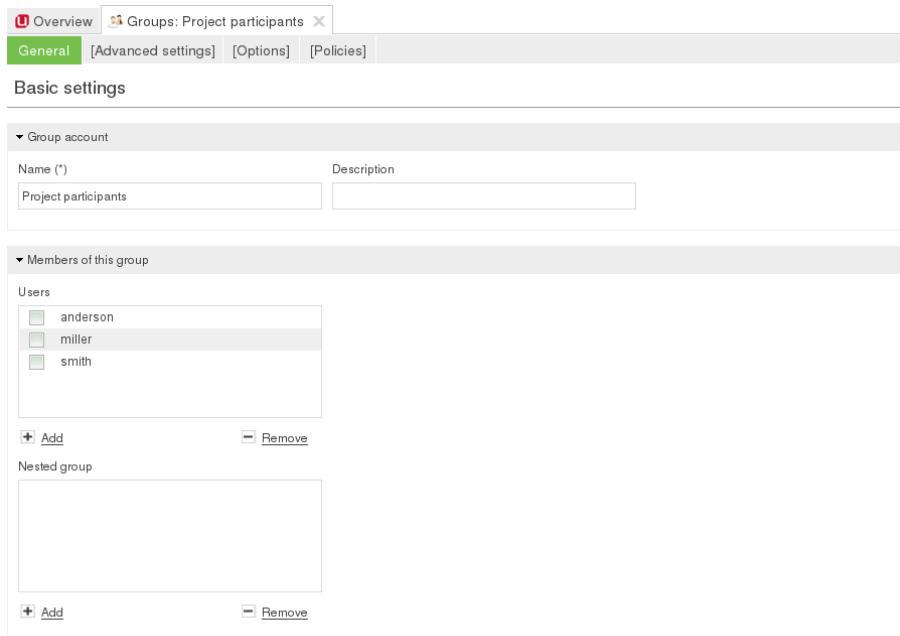
- A selection of groups can be assigned to a user in the user management (see Section 5.1)
- A selection of users can be assigned to a group in the group management (see Section 6.1)

6.1. Managing groups in the Univention Management Console

[Feedback](#) 

Groups are managed in the module *Groups* of the Univention Management Console.

Figure 6.1. Creating a group in UMC



The screenshot shows the 'Groups' module in the Univention Management Console. At the top, there's a navigation bar with tabs: 'Overview' (selected), 'Groups: Project participants' (with a close button), and 'General' (selected), along with other tabs: 'Advanced settings', 'Options', and 'Policies'. Below the tabs, the 'Basic settings' section is visible. It contains a 'Group account' panel with a 'Name (*)' field containing 'Project participants' and a 'Description' field. Underneath, the 'Members of this group' section shows a list of users: 'anderson', 'miller', and 'smith'. There are 'Add' and 'Remove' buttons for managing these members. Below this, a 'Nested group' section is shown, which is currently empty. There are also 'Add' and 'Remove' buttons for nested groups.

General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

Table 6.1. '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. 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 <code>directory/manager/user_group/uniqueness</code> is set to <code>false</code> , this check is removed.
Description	A description of the group can be entered here.
Users	This tab can be used for accepting users as members of the group.
Membership of other Groups	On this tab, other groups can be added as members of the current group (groups in groups).

Table 6.2. 'Advanced settings' tab

Attribute	Description
Mail	These options define a mail group and are documented in the Section 12.3.4.
Host members	This field can be used for accepting computers as members of the group.
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 grey. 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. The RID cannot be subsequently changed. When editing the group, the group ID will be represented in grey. The RIDs below 1000 are reserved for standard groups and other special objects. When Samba 4 is used, the RID is generated by Samba and cannot be specified.
Windows -> Samba group type	Three types of Windows groups can be distinguished: <ul style="list-style-type: none">• <i>Domain Groups</i> are known across the domain. This is the default group type.

Attribute	Description
	<ul style="list-style-type: none"> • <i>Local groups</i> are relevant on Windows servers exclusively. 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 NT domain, local groups in UCS can be handled in the same way as global groups. • <i>Well-Known Group</i>: This type contains groups pre-configured by Samba servers or by Windows servers, which can be available across the domain or locally restricted, e.g. <i>Adminusers</i>, <i>Printer Admins</i>, etc..
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 5.1.

Table 6.3. '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 <i>sambaGroupMapping</i> .
Posix group	This checkbox indicates whether the group contains the object class <i>posixGroup</i> .

6.2. 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 6.3) and is thus transparent for applications.

6.3. Local group cache

[Feedback](#)

The user and computer information retrieved from the LDAP is cached by the Name Server Cache Daemon, see Section 7.4.9.

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/extrousers/group` file by the `/usr/lib/univention-pam/ldap-group-to-file.py` script and read from there by the NSS module.

Local group cache

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 7.4.8.2) 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 verifies whether the group members are still present in the LDAP directory. If only the Univention Management Console is used for user management, this additional check is not necessary and can be disabled by setting the Univention Configuration Registry variable `nss/group/cachefile/check_member` to *false*.

Chapter 7. Computer management

7.1. Management of computer accounts in the Univention Management Console	82
7.1.1. Integration of Ubuntu clients	85
7.2. Configuration of hardware and drivers	85
7.2.1. Available kernel variants	85
7.2.2. Hardware drivers / kernel modules	86
7.2.3. GRUB boot manager	86
7.2.4. Network configuration	87
7.2.4.1. Network interfaces	87
7.2.4.2. Configuring proxy access	89
7.2.5. Configuration of the monitor settings	89
7.2.6. Mounting NFS shares	90
7.2.7. Collection of list of supported hardware	90
7.3. Administration of local system configuration with Univention Configuration Registry	90
7.3.1. Introduction	90
7.3.2. Using the Univention Management Console web interface	92
7.3.3. Using the command line front end	92
7.3.3.1. Querying a UCR variable	92
7.3.4. Policy-based configuration of UCR variables	94
7.3.5. Modifying UCR templates	94
7.3.5.1. Referencing of UCR variables in templates	95
7.3.5.2. Integration of inline Python code in templates	95
7.4. Basic system services	95
7.4.1. Administrative access with the root account	96
7.4.2. Configuration of language and keyboard settings	96
7.4.3. Starting/stopping system services / configuration of automatic startup	96
7.4.4. Authentication / PAM	97
7.4.4.1. Limiting authentication to selected users	97
7.4.5. Configuration of the LDAP server in use	98
7.4.6. Configuration of the print server in use	98
7.4.7. Logging/retrieval of system messages and system status	98
7.4.7.1. Log files	98
7.4.7.2. Logging the system status	98
7.4.7.3. Querying system statistics in the Univention Management Console	99
7.4.7.4. Process overview in the Univention Management Console	99
7.4.8. Executing recurring actions with Cron	100
7.4.8.1. Hourly/daily/weekly/monthly execution of scripts	100
7.4.8.2. Defining local cron jobs in /etc/cron.d	100
7.4.8.3. Defining cron jobs in Univention Configuration Registry	100
7.4.9. Nameserver cache daemon	101
7.4.10. SSH login to systems	101
7.4.11. Configuring the time zone / time synchronisation	102
7.5. Software deployment	102
7.5.1. Introduction	102
7.5.2. Differentiation of update variants / UCS versions	103
7.5.3. Univention App Center	103
7.5.4. Configuration of the repository server for updates and package installations	105
7.5.4.1. Configuration via the Univention Management Console	106
7.5.4.2. Configuration via Univention Configuration Registry	106
7.5.4.3. Policy-based configuration of the repository server	106
7.5.4.4. Creating and updating a local repository	106
7.5.5. Updates of UCS systems	107

7.5.5.1. Update strategy in environments with more than one UCS system	107
7.5.5.2. Updating individual systems via the Univention Management Console	107
7.5.5.3. Updating individual systems via the command line	108
7.5.5.4. Updating systems via a policy	109
7.5.5.5. Postprocessing of release updates	109
7.5.5.6. Troubleshooting in case of update problems	109
7.5.6. Installation of further software	109
7.5.6.1. Role-based software selection in the Univention Management Console	110
7.5.6.2. Installation/removal of individual packages in the Univention Management Con- sole	110
7.5.6.3. Installation/removal of individual packages in the command line	111
7.5.6.4. Policy-based installation/uninstallation of individual packages via package lists....	111
7.5.7. Specification of an update point using the package maintenance policy	112
7.5.8. Integration of components not provided through the Univention App Center	112
7.5.8.1. Integration of repository components via the Univention Management Console....	113
7.5.8.2. Integration of repository components via Univention Configuration Registry	114
7.5.9. Central monitoring of software installation statuses with the software monitor	114

7.1. Management of computer accounts in the 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.

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 **Computers** module of the Univention Management Console.

Table 7.1. 'General' tab

Attribute	Description
Name	<p>The name for the host should be entered in this input field.</p> <p>To guarantee compatibility with different operating systems and services, computer names should only contain the lowercase letters <i>a</i> to <i>z</i>, 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.</p> <p>Microsoft Windows accepts computer names with a maximum of 15 characters, so as a rule computer names should be limited to 15 characters if there is any chance that Microsoft Windows will be used.</p>

Attribute	Description
	After creation, the computer name can only be changed for the system roles <i>Windows Workstation/Server</i> , <i>Mac OS X Client</i> and <i>IP managed client</i> .
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 9.1.
MAC address	The MAC address of the computer can be entered here e.g., <i>2e:44:56:3f:12:32</i> . If the computer is to receive a DHCP entry, the entry of the MAC address is essential.
IP address	<p>Fixed IP addresses for the host can be given here. Further information on the IP configuration can be found in Section 9.1.</p> <p>If a network was selected on the General tab, the IP address assigned to the host from the network will be shown here automatically.</p> <p>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 7.2.4.</p> <p>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.</p>
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 9.1.
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 9.1.
DHCP service	<p>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 9.1.</p> <p>During assignment, it must be ensured that the DHCP servers of the DHCP service object are responsible for the physical network.</p> <p>If a network is selected on the General tab an appropriate entry for the network will be added automatically. It can be adapted subsequently.</p>

Table 7.2. '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.

Attribute	Description
	The same password must then also be entered locally on the computer in the <code>/etc/machine.secret</code> 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 <i>DC Backup Hosts</i> , for a DC slave <i>DC Slave Hosts</i> and for member services, managed clients and mobile clients <i>Computers</i> .

Table 7.3. '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 <code>/dev/null</code>
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 <code>/bin/sh</code> .

Table 7.4. '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 7.5. 'Deployment' tab (advanced settings)

This tab is used for the Univention Net Installer, see [ext-doc-inst].

Table 7.6. '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 7.7. 'Groups' tab (advanced settings)

The computer can be added into different groups in this tab.

Table 7.8. 'Nagios services' tab (advanced settings)

This tab is used to specify which Nagios tests should be performed for this computer, see Section 13.3.3.

Table 7.9. 'Nagios notification' tab (advanced settings)

This tab is used to specify which users should be informed if Nagios tests should fail, see Section 13.3.3.

Table 7.10. '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 14.

Table 7.11. '(Options)' tab

Attribute	Description
This tab allows to disable LDAP object classes for host objects. The entry fields for attributes of disabled object classes are no longer shown. Not all object classes can be modified subsequently.	
Kerberos principal	If this checkbox is not selected the host does not receive the <i>krb5Principal</i> and <i>krb5KDCEntry</i> object classes.
Posix account	If this checkbox is not selected the host does not receive the <i>posixAccount</i> 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 <i>sambaSamAccount</i> object class.

7.1.1. Integration of Ubuntu clients

[Feedback](#)

Ubuntu clients can be managed in the Univention Management Console with their own system role. The network properties for DNS/DHCP can also be managed via the 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 Univention Wiki [ubuntu-integration].

7.2. Configuration of hardware and drivers

[Feedback](#)

7.2.1. Available kernel variants

[Feedback](#)

The standard kernel in UCS 3.1 is based on the Linux kernel 3.2. 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

7.2.2. 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 7.2.3) 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 recognised 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 recognised (which can occur if no respective hardware IDs are registered or hardware is employed which cannot be recognised 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.

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 standardised 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.

7.2.3. 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.

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 colour 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 http://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/xenhyp` 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.1st` to boot paravirtualized Xen systems. This file is generated automatically if it does not yet exist. This behaviour can be deactivated by setting the Univention Configuration Registry variable `grub/generate-menu-1st` to `no`.

7.2.4. Network configuration

[Feedback](#)

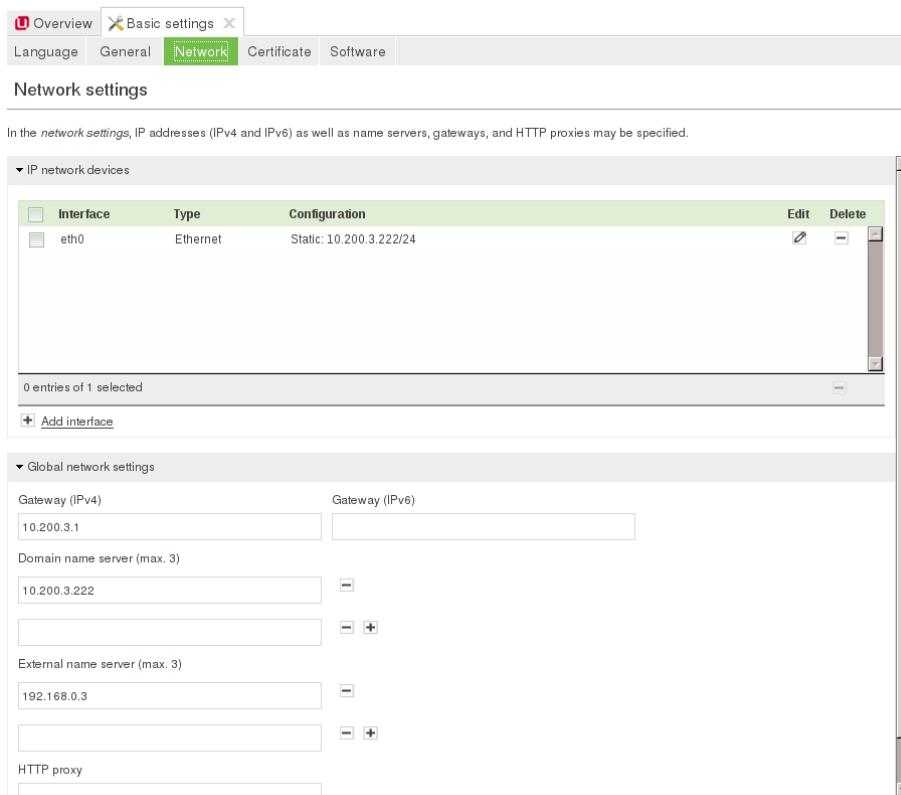
7.2.4.1. Network interfaces

[Feedback](#)

The configuration of network interfaces can be adjusted in the Univention Management Console under **Basic settings -> Network**.

The configuration is saved in Univention Configuration Registry variables, which can also be set directly. These variables are listed in brackets in the individual sections.

Figure 7.1. 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., 192.168.0.10), whereas IPv6 addresses are four times as long and typically written in hexadecimal form (e.g., 2222:0DFE:FE29:DE27:0000:0000:0000:0000).

7.2.4.1.1. Configuration of IPv4 addresses

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`.

7.2.4.1.2. Configuration of IPv6 addresses

The IPv6 address can be configured in two ways: Stateless address autoconfiguration (SLAAC) is employed in the **Autoconfiguration (SLAAS)** 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.

(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`)

7.2.4.1.3. Configuring the name servers

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`,

7.2.4.1.4. Configuration of bridges/bonding/VLANs

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.

Bridges, bondings and VLANs are displayed in the Univention Management Console, but the configuration of such setups can only be performed through Univention Configuration Registry variables. The setup is explained in the extended computer documentation [ext-doc-computers].

7.2.4.2. Configuring proxy access

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://192.168.1.100`. The proxy port can be specified in the proxy URL using a colon, e.g., `http://192.168.1.100:3128`. If the proxy requires authentication for the accessing user, this can be provided in the form `http://username:password@192.168.1.100`.

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 `univention.de` also applies for `apt.univention.de`.

7.2.5. Configuration of the monitor settings

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.

Managed and mobile clients (only installable with Univention Corporate Desktop) and thin clients (only installable with Univention Thin Client Services) can also be manually configured with a **Display** policy. The settings can be set with a Univention Configuration Registry policy on the remaining system roles.

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 brackets:

- **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`).

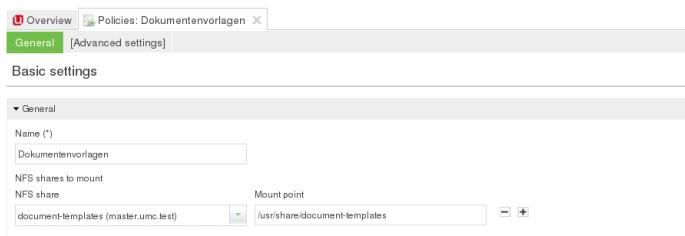
- **Resolution of secondary display** defines the screen resolution of a second monitor, if present. This combines with the primary monitor to display a shared screen area (`xorg/resolution/secondary`).
- The **Position of secondary display** menu specifies the relative position of the secondary monitor 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 colour colour depth) (`xorg/screen/DefaultDepth`).

7.2.6. Mounting NFS shares

[Feedback](#)

The **NFS mounts** policy of the computer management of the Univention Management Console 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 7.2. Mounting a NFS share



7.2.7. Collection of list of supported hardware

[Feedback](#)

Univention maintains a list of the hardware [hardwarelist] which is compatible with UCS and in use by customers. The information processed for this are 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.

7.3. Administration of local system configuration with Univention Configuration Registry

[Feedback](#)

7.3.1. 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 the 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 7.3.5.

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 than 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 7.3.4.

7.3.2. Using the Univention Management Console web interface

[Feedback](#) 

The **Univention Configuration Registry** module of the Univention Management Console 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 stylised pen edits the setting of a variable. The icon with the stylised minus sign allows the deletion of a variable.

Some central variables - e.g., the domain name - cannot be changed directly in Univention Configuration Registry. Instead they can only be changed indirectly via the basic settings in the Univention Management Console. These variables can be shown in grey. It is still possible to edit them via the command line front-end.

7.3.3. 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.

7.3.3.1. 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
```

7.3.3.1.1. 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=192.168.0.2 \
    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?192.168.0.2
```

7.3.3.1.2. 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 <http://docs.python.org/library/re.html>.

7.3.3.1.3. 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
```

7.3.3.1.4. 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.

7.3.3.1.5. 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.

Policy-based configuration of UCR variables

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:

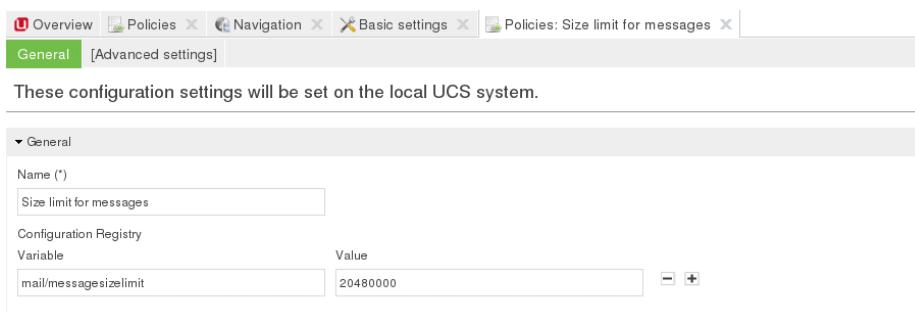
```
# eval "$(univention-config-registry shell ldap/server/name)"
# echo "$ldap_server_name"
master.firma.de
```

7.3.4. 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 than one computer. The **Univention Configuration Registry** policy managed in the **Policies** module of the Univention Management Console can be used to compile variables and apply them on more than one computer.

Figure 7.3. 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 4.5.2). Note that the evaluation of configured values differs from other policies: the values are not forwarded directly to the computer, 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.

7.3.5. 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 resaved 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.

7.3.5.1. 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 7.3.5.2).

7.3.5.2. 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 realise 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 7.3.5.1), 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.

7.4. 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.

7.4.1. Administrative access with the root account

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.

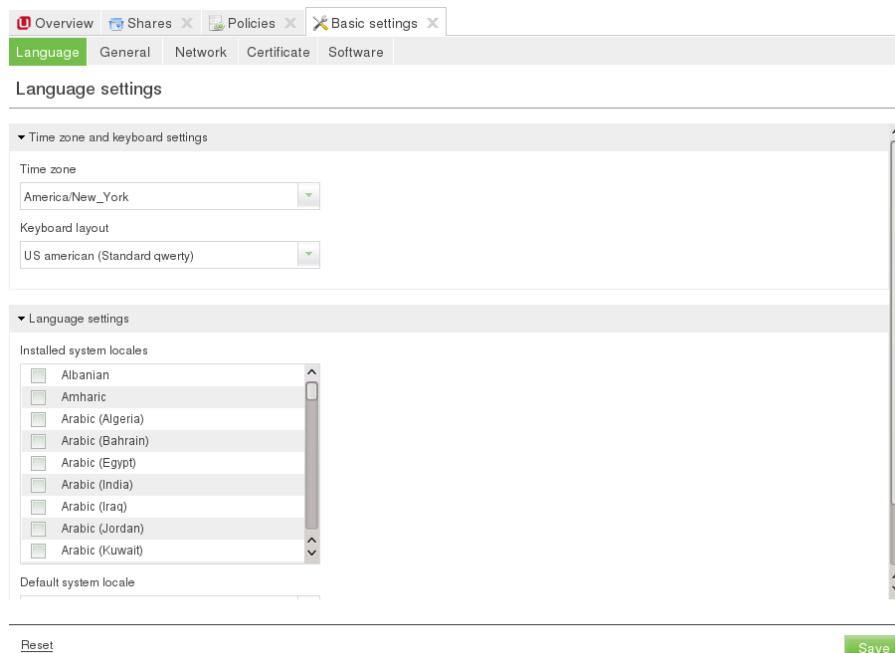
The password for the root user can be changed via the Univention Management Console under **Basic settings -> General -> Root password**. Alternatively, the `passwd` command can be run in the command line.

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. To avoid subsequent errors by misspelling, the password has to be entered twice.

7.4.2. Configuration of language and keyboard settings

In Linux, localisation 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 internationalised programs. The installed locales can be changed in the Univention Management Console under **Basic settings -> Language -> Installed system locales**. The standard locale is set under **Default system locale**.

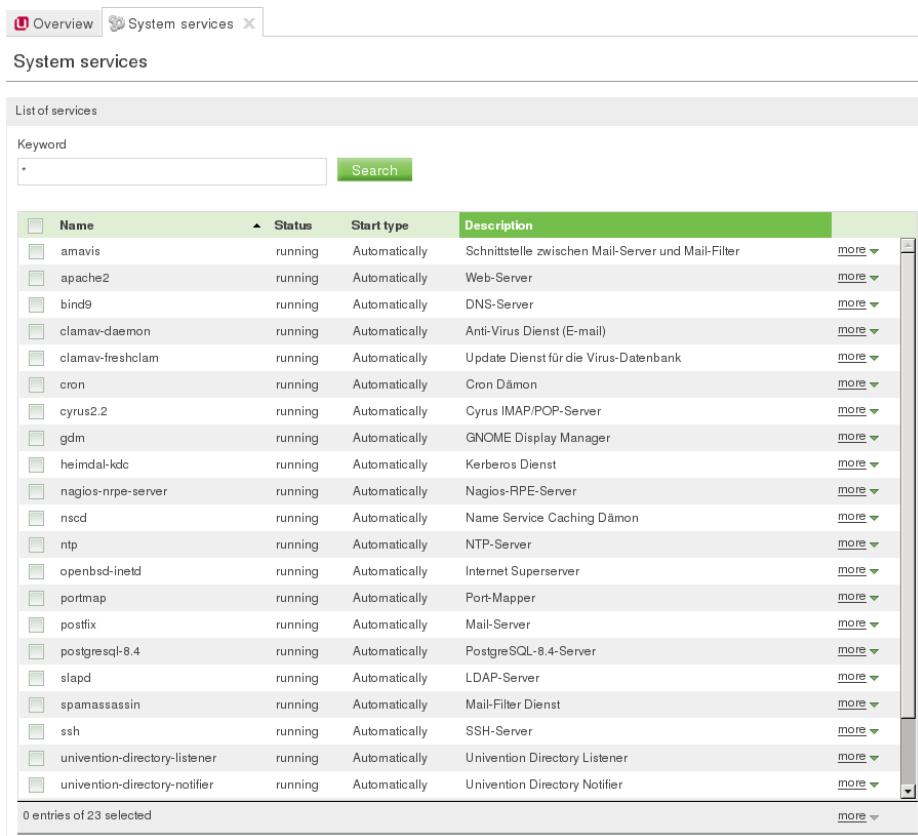
Figure 7.4. Configuring the language settings



The **Keyboard layout** in the menu entry **Time zone and keyboard settings** is applied during local logins to the system.

7.4.3. Starting/stopping system services / configuration of automatic startup

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 7.5. Overview of system services


Name	Status	Start type	Description	more
amavis	running	Automatically	Schnittstelle zwischen Mail-Server und Mail-Filter	more
apache2	running	Automatically	Web-Server	more
bind9	running	Automatically	DNS-Server	more
clamav-daemon	running	Automatically	Anti-Virus Dienst (E-mail)	more
clamav-freshclam	running	Automatically	Update Dienst für die Virus-Datenbank	more
cron	running	Automatically	Cron Dämon	more
cyrus2.2	running	Automatically	Cyrus IMAP/POP-Server	more
gdm	running	Automatically	GNOME Display Manager	more
heimdal-kdc	running	Automatically	Kerberos Dienst	more
nagios-nrpe-server	running	Automatically	Nagios-RPE-Server	more
nsqd	running	Automatically	Name Service Caching Dämon	more
ntp	running	Automatically	NTP-Server	more
openbsd-inetd	running	Automatically	Internet Superserver	more
portmap	running	Automatically	Port-Mapper	more
postfix	running	Automatically	Mail-Server	more
postgresql-8.4	running	Automatically	PostgreSQL-8.4-Server	more
slapd	running	Automatically	LDAP-Server	more
spamassassin	running	Automatically	Mail-Filter Dienst	more
ssh	running	Automatically	SSH-Server	more
univention-directory-listener	running	Automatically	Univention Directory Listener	more
univention-directory-notifier	running	Automatically	Univention Directory Notifier	more

0 entries of 23 selected [more](#)

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.

[Feedback](#)

7.4.4. Authentication / PAM

[Feedback](#)

Authentication services in Univention Corporate Server are realised 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.

[Feedback](#)

7.4.4.1. Limiting authentication to selected users

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 authorised 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 GDM (`gdm`), login on a tty (`login`), rlogin (`rlogin`), PPP (`ppp`) and other services (`other`). An example for SSH:

```
auth/sshd/group/Administrators: yes
```

Configuration of the LDAP server in use

```
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
```

7.4.5. 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 computer management of the Univention Management Console. The order of the servers determines the order of the computer's requests to the server if a LDAP server cannot be reached.

7.4.6. 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 computer management of the Univention Management Console.

7.4.7. Logging/retrieval of system messages and system status

[Feedback](#) 

7.4.7.1. 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.

7.4.7.2. 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 indices 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.

7.4.7.3. Querying system statistics in the Univention Management Console

[Feedback](#) 

The UMC module **Statistics** displays the utilisation 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 utilisation of the main memory in percent
- The processor utilisation of the system
- The number of terminal server sessions active
- The utilisation of the swap file

7.4.7.4. Process overview in the Univention Management Console

[Feedback](#) 

The UMC system 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 utilisation 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.

7.4.8. 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.

7.4.8.1. 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.

7.4.8.2. 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
```

7.4.8.3. 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 7.4.8.2) (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.

7.4.9. Nameserver 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 6.3.

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 7.12. Default size of the hash table

Variable	Default size of the hash table
<code>nscd/hosts/size</code>	6007
<code>nscd/passwd/size</code>	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`.

7.4.10. SSH login to systems

[Feedback](#) 

When installing a UCS system, an SSH server is also installed per preselection. SSH is used for realising 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`.

7.4.11. Configuring the time zone / time synchronisation

[Feedback](#) 

The time zone in which a system is located can be changed in the Univention Management Console under **Basic settings -> Language -> 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 Domain Controller Master 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 synchronisation 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. This setting should be set on Samba 4 domain controllers.

7.5. Software deployment

[Feedback](#) 

7.5.1. 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 an online update.

For larger installations, there is the possibility of establishing a local repository server from which all further updates can then be performed. 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 (see Section 7.5.4).

UCS systems can be updated via the Univention Management Console, a command line tool and policy-based. This is described in the Section 7.5.5.

The UCS software deployment does not support the updating of Microsoft Windows systems. An additional Windows software distribution is required for this.

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

The functional scope of a UCS installation can be expanded by integrating further repository components, see Section 7.5.8. These components are provided both by Univention (e.g., for the UCS-based UCS DVS) and by third parties.

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 7.5.6.

The installation of version and errata updates can be automated via policies, see Section 7.5.7

The software monitor provides a tool with which all package installations statuses can be centrally stored in a database, see Section 7.5.9.

[Feedback](#) 

7.5.2. Differentiation of update variants / UCS versions

Four different types of UCS updates are differentiated:

- *Major releases* appear approximately every 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 6-8 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 behaviour prove practical or unavoidable, this will be noted in the release notes when the new version is published.
- *Patchlevel releases* include fewer changes than a minor release. The focus here is exclusively on correcting errors. Compatibility with the previous version in terms of the functioning, interfaces and operation is maintained insofar as a change cannot be avoided without excessive efforts or corrects a critical error. Should a change, however, prove unavoidable, this will be noted in the release notes when the new version is published. New patch level versions are released approximately every two months.
- Alongside the planned, long-term maintenance and on-going development of UCS via major, minor and patch level versions, Univention also publishes *errata updates* as necessary. Errata updates include the corrections for security vulnerabilities or critical program errors which need to be resolved quickly prior to the release of the next patch level update. A list of the errata updates can be found at <http://errata.univention.de/>. Errata updates can also be provided for components.

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 3.2-1 thus refers to the first patch level update for the second minor update for the major release UCS 3.x.

Errata updates always refer to certain minor releases, e.g., for UCS 3.0. 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.

An overview of all the errata updates can be found under <http://errata.univention.de/>.

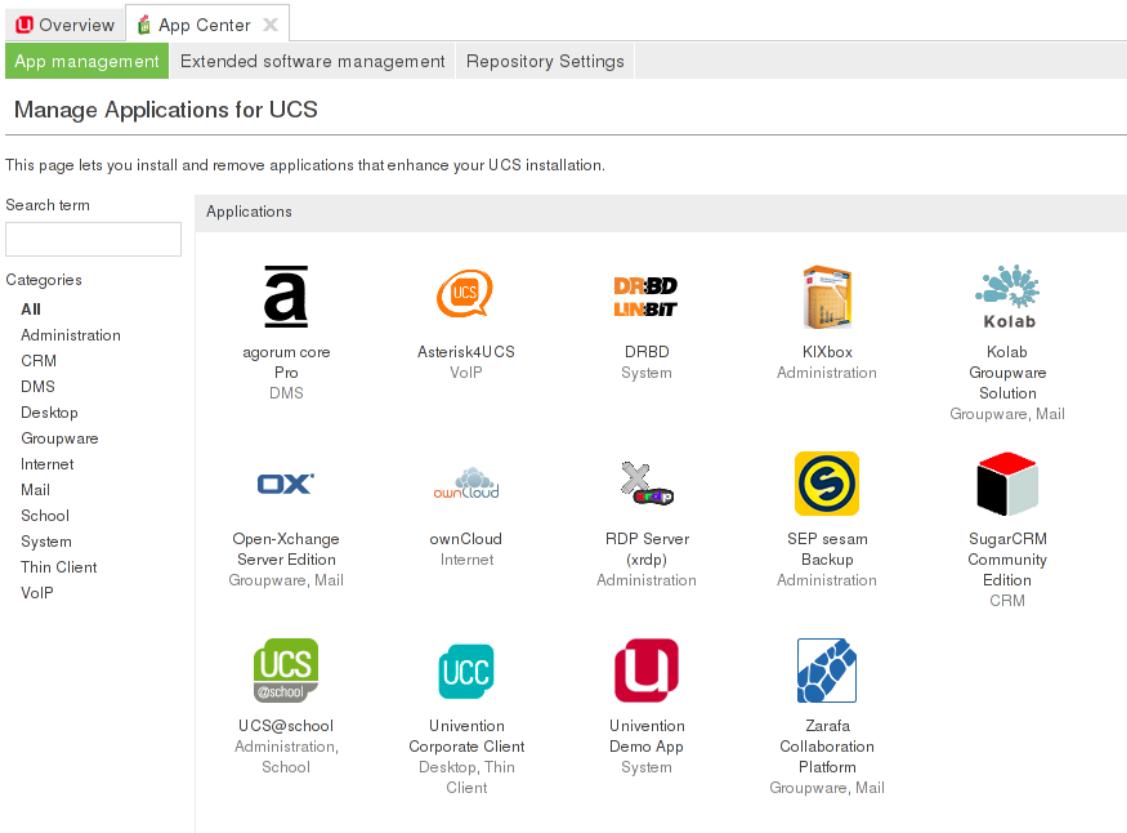
A changelog document is published for every release update listing the updated packages, information on error corrections and new functions and references to the Univention Bugzilla.

[Feedback](#) 

7.5.3. Univention App Center

The Univention App Center allows simple integration of software components in a UCS installation. The applications are provided both by third parties and by Univention itself (e.g., ucs@School). Licenses, maintenance and support for the applications are purchased from the respective manufacturer and not via the Univention App Center.

Figure 7.6. Overview of applications available in the App Center



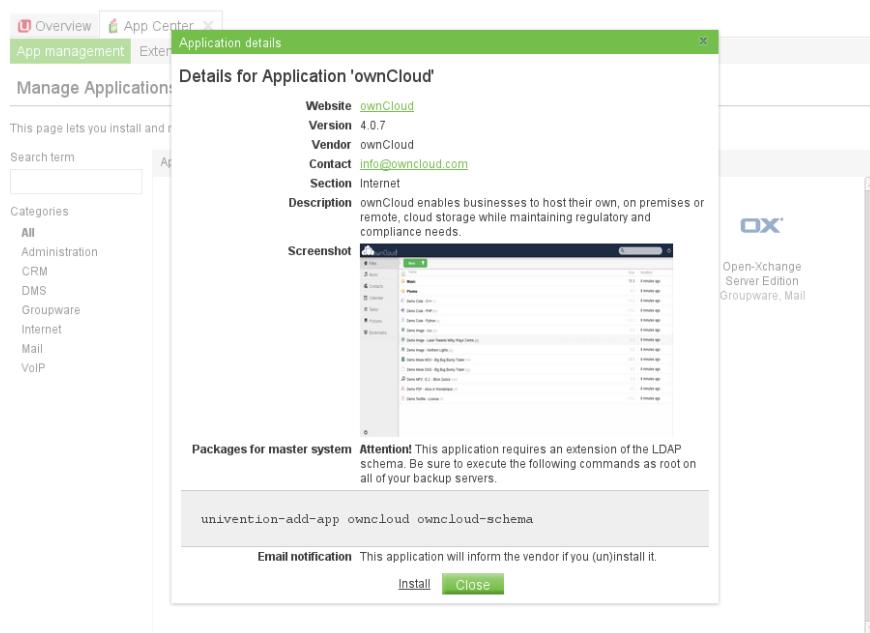
The screenshot shows the Univention App Center interface. At the top, there is a navigation bar with tabs: "Overview" (selected), "App Center" (with a close button), "App management" (highlighted in green), "Extended software management", and "Repository Settings". Below the navigation bar, the title "Manage Applications for UCS" is displayed. A sub-instruction states: "This page lets you install and remove applications that enhance your UCS installation." On the left, there is a sidebar with a "Search term" input field and a "Categories" section containing a list of software components categorized under "All": Administration, CRM, DMS, Desktop, Groupware, Internet, Mail, School, System, Thin Client, and VoIP. The main area is titled "Applications" and contains a grid of 12 application icons, each with its name, description, and category. The applications listed are:

Icon	Name	Description	Category
	agorum core Pro DMS		
	Asterisk4UCS VoIP		
	DRBD System		
	KIXbox Administration		
	Kolab Groupware Solution	Groupware, Mail	
	Open-Xchange Server Edition Groupware, Mail	Groupware, Mail	
	ownCloud Internet	Internet	
	RDP Server (xrdp) Administration	Administration	
	SEP sesam Backup Administration	Administration	
	SugarCRM Community Edition CRM	CRM	
	UCS@school Administration, School	Administration, School	
	Univention Corporate Client Desktop, Thin Client	Desktop, Thin Client	
	Univention Demo App System		
	Zarafa Collaboration Platform Groupware, Mail	Groupware, Mail	

The Univention App Center can be opened via the UMC module *App Center*. The available software components are listed under **Applications**. **Search term** can be used to filter the list of displayed applications. The applications can also be sorted using the **Categories**.

If you click on one of the displayed applications, further details on it are shown (e.g., description, manufacturer, contact information and a screenshot). The **Email notification** field shows whether the manufacturer of the software component is notified when it is installed/uninstalled.

Figure 7.7. Details for an application in the App Center



Installation of most of the applications in the Univention App Center requires a personalised license key. UCS license keys granted for UCS versions older than 3.1 need to be converted. If an affected application is installed, a window opens where you can start the conversion process: 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.

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.

Many 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.

Following installation of an application, two new options are shown when clicking on the icon of an application: **Uninstall** removes an application and clicking on **Open** displays an overview module containing detailed information and a link to the documentation as well as the installed version. Here one can also uninstall the application or - if a more up-to-date version is available - perform an update. An overview of the existing applications 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.

[Feedback](#)

7.5.4. Configuration of the repository server for updates and package installations

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 and 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, e.g., other mail servers than Postfix can be found under *unmaintained*. 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.

7.5.4.1. Configuration via the Univention Management Console

[Feedback](#) 

The repository settings can be adjusted in the **App Center** module of the Univention Management Console. The **Repository server** and the use of the maintained and unmaintained sections can be specified in the **Repository Settings** tab.

7.5.4.2. 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`.

7.5.4.3. 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.

General information on policy management can be found in Section 4.5.

7.5.4.4. 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 synchronising local repositories, which means, for example, a main repository is maintained at the company headquarters and then synchronised 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.

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 synchronised 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`.

7.5.5. Updates of UCS systems

[Feedback](#) 

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

7.5.5.1. 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.

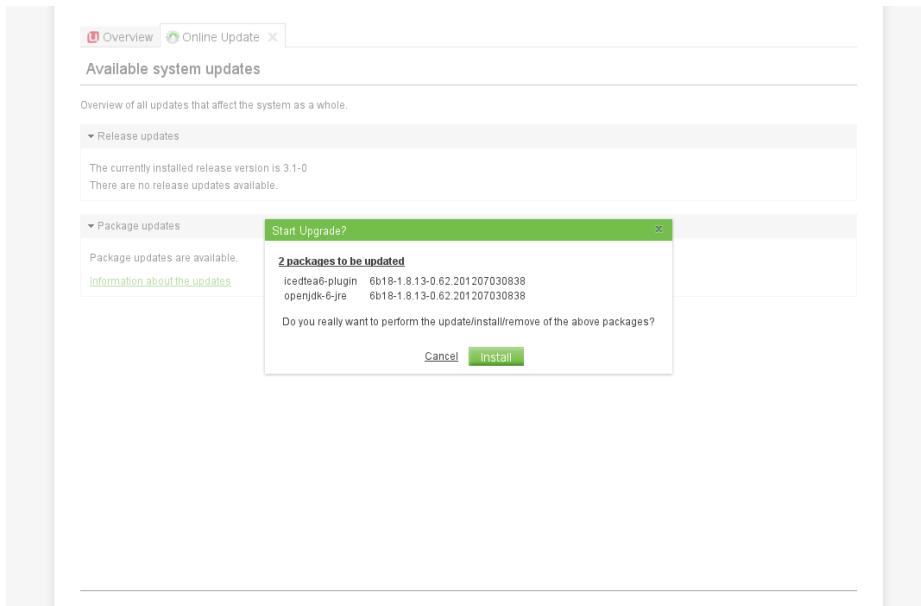
It should be noted, that components marked as critical can block an upgrade to the next minor or major release. More information can be found in Section 7.5.8.

7.5.5.2. Updating individual systems via the Univention Management Console

[Feedback](#) 

The **Online Update** module allows the installation of release updates and errata updates.

Figure Figure 7.8 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 7.8. 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`

7.5.5.3. 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 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 7.5.4.4), 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.

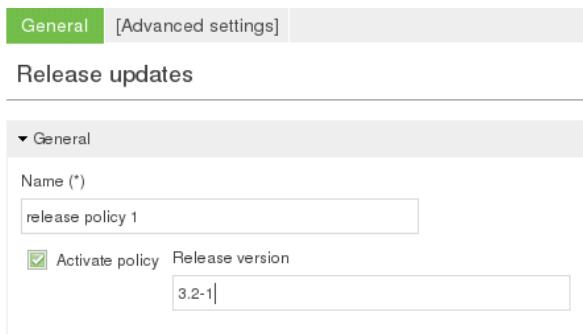
The messages created during the update are written to the file `/var/log/univention/updater.log`

7.5.5.4. Updating systems via a policy

An update for more than one computer can be configured with a **Release** policy in the computer/domain management of the Univention Management Console.

General information on policy management can be found in Section 4.5.

Figure 7.9. Updating UCS systems using a release policy



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

The **Release version** input field includes the version number up to which the system should be updated, e.g., 3.2-1. 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 **Paketpflege** policy (see Section 7.5.7).

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

7.5.5.5. Postprocessing of release updates

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).

7.5.5.6. Troubleshooting in case of update problems

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.

7.5.6. Installation of further software

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.

Installation of further software

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 7.5.4).

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.

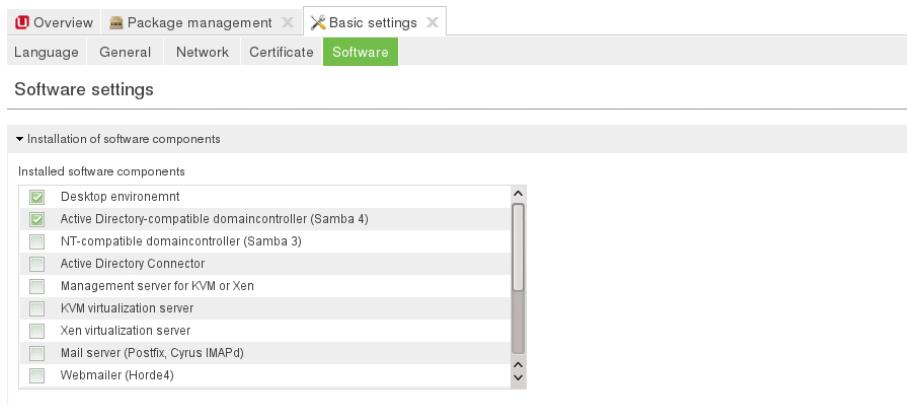
There are several possibilities for installing further packages subsequently on an installed system:

7.5.6.1. Role-based software selection in the Univention Management Console

[Feedback](#) 

A module is provided under **Basic settings -> Software** in the Univention Management Console with which the role-based software selection also offered in the Univention Installer can be subsequently adjusted. Software components can be installed or removed by activating/deactivating the selection box in front of the components and then clicking on save.

Figure 7.10. Software selection



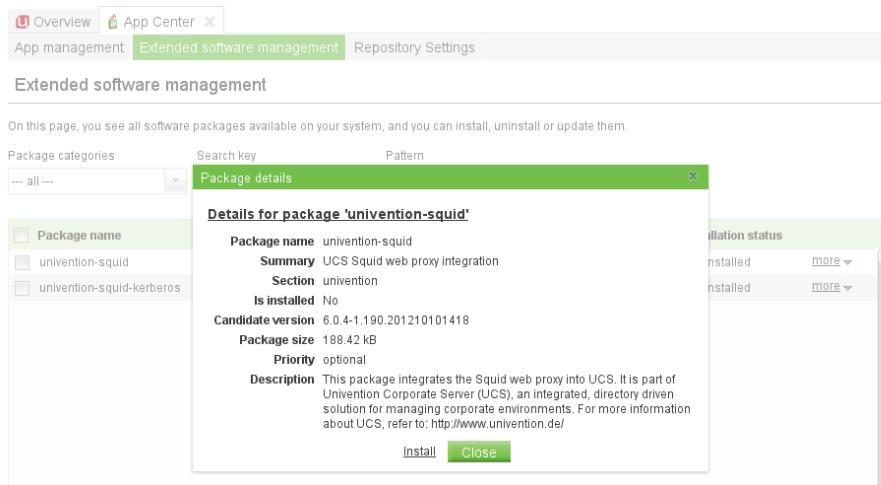
A list of the available software selection can be found in Section 2.12.

7.5.6.2. Installation/removal of individual packages in the Univention Management Console

[Feedback](#) 

The Univention Management Console module **App Center** (tab: **Extended software management**) can be used to install and uninstall individual software packages.

Figure 7.11. Installing the package univention-squid in the 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.

[Feedback](#)

7.5.6.3. Installation/removal of individual packages in the command line

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
```

[Feedback](#)

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

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 *Policies* module of the Univention Management Console with the **Policy: Packages + system role**.

Table 7.13. 'General' tab

Attribute	Description
Name	An unambiguous name for this package list, e.g., <i>mail server</i> .
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 7.5.7).

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

7.5.7. Specification of an update point using the package maintenance policy

[Feedback](#) 

A **Maintenance** policy in the computer/domain management of the Univention Management Console 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 7.5.5.4) and, if applicable, installation.
- Installation/uninstallation of package lists (see Section 7.5.6.4)
- Installation of available errata updates

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

General information on policy management can be found in Section 4.5.

Table 7.14. 'General' tab

Attribute	Description
System startup	If this option is activated, the update steps are performed when the computer is started up.
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 <i>Month</i> , <i>Day of week</i> , <i>Day</i> , <i>Hour</i> and <i>Minute</i> 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.

7.5.8. Integration of components not provided through the Univention App Center

[Feedback](#) 

In the default setting, a repository only includes the packages of the standard UCS scope. Further components can be added both from Univention - e.g., UCS@school - or from third parties such as groupware manufacturers. Additional components are generally integrated via the Univention App Center (see Section 7.5.3). The following steps are only necessary for components which are not available from the Univention App Center.

Additional components can be registered via the Univention Management Console or in Univention Configuration Registry.

Components can be versioned. This ensures that only components are installed that are compatible with a UCS version. The version compatibility is automatically ensured during installation via the Univention App Center.

empty or unset

All versions of the same major number will be used. If for example UCS-2.3 is installed, all repositories of the component with version numbers 2.0, 2.1, 2.2 and 2.3 will be used if available.

current

current Using the keyword *current* will likewise include all versions of the same major version. Additionally it will block all minor and major upgrades of the installed UCS system until the respective component is also available for the new release. Patch level and errata updates are not affected. If for example UCS-2.3 is currently installed and UCS-2.4 or UCS-3.0 is already available, the release update will be postponed until the component is also available for version 2.4 and 3.0 respectively.

major.minor

By specifying an explicit version number only the specified version of the component will be used. Release updates of the system will not be hindered by such components. Multiple versions can be given using commas as delimiters, for example 2.0,2.3.

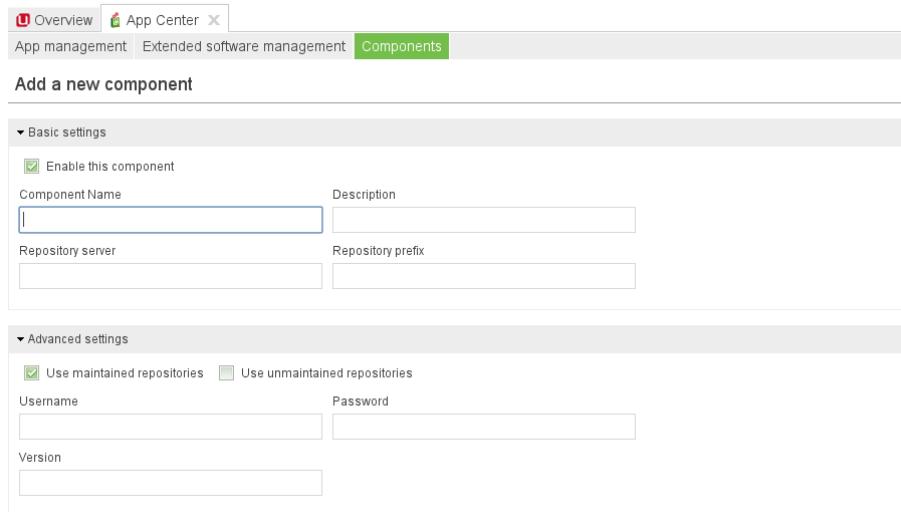
7.5.8.1. Integration of repository components via the Univention Management Console

[Feedback](#) 

A list of the integrated repository components can be found in the **Repository Settings** tab in the **App Center** module of the Univention Management Console. Applications which have been added via the Univention App Center are still listed here, but should be managed via the **App management** tab.

A further component can be set up with **Add**. The **Component Name** identifies the component on the repository server. A free text can be entered under **Description**, for example, for describing the functions of the component in more detail.

Figure 7.12. Configuring a repository component in UMC



The host name of the download server is to be entered in the input field **Repository server**, and, if necessary, an additional file path in **Repository prefix**.

A **Username** and **Password** can be configured for repository servers which require authentication.

A software component is only available once **Enable this component** has been activated.

A differentiation is also made for components between *maintained* and *unmaintained* components, see Section 7.5.4.

7.5.8.2. Integration of repository components via Univention Configuration Registry Feedback

The following Univention Configuration Registry variables can be used to register a repository component. It is also possible to activate further functions here which cannot be configured via the UMC module. *NAME* stands for the component's name:

`repository/online/component/NAME/server`

The repository server on which the components are available. If this variable is not set, the server from the Univention Configuration Registry variable `repository/online/server` uses.

`repository/online/component/NAME`

This variable must be set to *enabled* if the components are to be mounted.

`repository/online/component/NAME/localmirror`

This variable can be used to configure whether the component is mirrored locally. In combination with the Univention Configuration Registry variable `repository/online/component/NAME/server`, a configuration can be set up so that the component is mirrored, but not activated, or that it is activated, but not mirrored.

`repository/online/component/NAME/description`

A descriptive name for the repository.

`repository/online/component/NAME/prefix`

Defines the URL prefix which is used on the repository server. This variable is usually not set.

`repository/online/component/NAME/username`

If the repository server requires authentication, the user name can be entered in this variable.

`repository/online/component/NAME/password`

If the repository server requires authentication, the password can be entered in this variable.

`repository/online/component/NAME/version`

This variable controls the versions to include, see Section 7.5.8 for details.

`repository/online/component/NAME/defaultpackages`

A list of package names separated by blanks. The UMC module *App Center* offers the installation of this component if at least one of the packages is not installed. Specifying the package list eases the subsequent installation of components.

For example, to integrate the UCS@School component, the following variables must be set:

```
ucr set repository/online/component/ucsschool/description="UCS@School
packages" \
repository/online/component/ucsschool/server=updates.software-
univention.de \
repository/online/component/ucsschool=yes
```

7.5.9. 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.

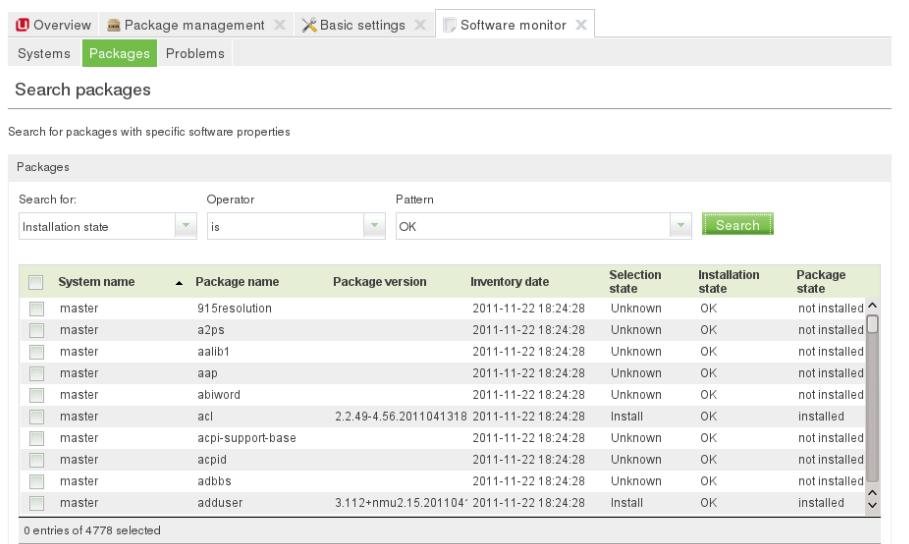
The software monitor can be set up by selecting the *Software installation monitor* component in the Univention Installer. Alternatively, the ***univention-pkgdb*** package can be installed.

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 the Univention Management Console and can be accessed via the **Software monitor** module. The following functions are available:

- *Systems* allows to search for UCS systems, which are tracked by the software monitor. 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. The following gives an overview of the 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 cancelled in different phases.
- *Problems* allows automatic identification of installation problems:
 - The system installation status can be compared with a given target release. This allows the efficient discovery of out-of-date installations in large environments.
 - Incompletely installed software packages can be identified.

Figure 7.13. Searching for packages in the software monitor



The screenshot shows the Univention Management Console interface for the Software monitor. The top navigation bar includes tabs for Overview, Package management, Basic settings, and Software monitor. The Software monitor tab is active. Below the tabs, there are three buttons: Systems, Packages (which is highlighted in green), and Problems. A search bar labeled "Search packages" is present. Underneath, a sub-header says "Search for packages with specific software properties". A search form is shown with fields for "Search for:" (containing "Installation state"), "Operator" (containing "is"), and "Pattern" (containing "OK"). A "Search" button is next to the pattern field. Below the search form is a table titled "Packages" with columns: System name, Package name, Package version, Inventory date, Selection state, Installation state, and Package state. The table lists several packages installed on a system named "master". At the bottom of the table, it says "0 entries of 4778 selected".

System name	Package name	Package version	Inventory date	Selection state	Installation state	Package state
master	915resolution		2011-11-22 18:24:28	Unknown	OK	not installed
master	a2ps		2011-11-22 18:24:28	Unknown	OK	not installed
master	aalib1		2011-11-22 18:24:28	Unknown	OK	not installed
master	aap		2011-11-22 18:24:28	Unknown	OK	not installed
master	abiword		2011-11-22 18:24:28	Unknown	OK	not installed
master	acl	2.2.49-4.56.2011041318	2011-11-22 18:24:28	Install	OK	installed
master	acpi-support-base		2011-11-22 18:24:28	Unknown	OK	not installed
master	acpid		2011-11-22 18:24:28	Unknown	OK	not installed
master	adbbs		2011-11-22 18:24:28	Unknown	OK	not installed
master	adduser	3.112+nmu2.15.201104	2011-11-22 18:24:28	Install	OK	installed

Central monitoring of software installation statuses with 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 8. Services for Windows

8.1. Introduction	117
8.2. Installation and components of a Samba domain	118
8.2.1. Installation and components of a Samba 3 domain	118
8.2.2. Installation and components of a Samba 4 domain	119
8.3. Services of a Samba domain	119
8.3.1. Authentication services	119
8.3.2. File services	119
8.3.3. Print services	120
8.3.4. Univention S4 connector	120
8.3.5. NetBIOS name service	120
8.3.6. Name resolution using WINS	120
8.4. Configuration and management of Windows desktops	121
8.4.1. Group policies	121
8.4.2. Logon scripts / NETLOGON share	121
8.4.2.1. Samba 3	121
8.4.2.2. Samba 4	122
8.4.3. Configuration of the file server for the home directory	122
8.4.3.1. Configuration with Samba 3	122
8.4.3.2. Configuration with Samba 4	122
8.4.4. Roaming profiles	122
8.4.4.1. Samba 3	122
8.4.4.2. Samba 4	123
8.4.5. Granting additional Windows privileges to users	123
8.5. UCS Active Directory Connector	123
8.5.1. Introduction	123
8.5.2. Setup of the UCS AD connector	124
8.5.2.1. Basic configuration of the connector	124
8.5.2.2. Importing the SSL certificate of the Active Directory	126
8.5.2.3. Setting up the password service on the AD system	127
8.5.2.4. Starting/Stopping the Active Directory connector	128
8.5.2.5. Functional test of basic settings	128
8.5.3. Additional tools / Debugging connector problems	128
8.5.3.1. univention-adsearch	128
8.5.3.2. univention-connector-list-rejected	128
8.5.3.3. Logfiles	129
8.5.4. Details on preconfigured synchronisation	129
8.5.4.1. Containers and organisational units	129
8.5.4.2. Groups	129
8.5.4.3. Users	130
8.6. Trust relationships	131
8.6.1. Windows domain trusts Samba 3 domain	131
8.6.2. Samba 3 domain trusts Windows domain	131

8.1. Introduction

[Feedback](#) 

For the purposes of Windows systems, UCS can assume the tasks of Windows server systems:

- Domain controller function
- File services
- Authentication services

- Print services

All these services are provided by Samba in UCS. Two different Samba versions can be employed in UCS:

- *Samba 3* implements domain services based on the domain technology of Microsoft Windows NT. Samba 3 is the current, stable and tried-and-tested main release series of the Samba project and has been integrated in UCS for many years.
- *Samba 4* is the next generation of the Samba suite. The most important innovation of Samba 4 is the support of domain, directory and authentication services which are compatible with Microsoft Active Directory. This means that Active Directory-compatible Windows domains can be constructed with Samba 4. These also allow the use of the tools provided by Microsoft for the management of users or group policies (GPOs) for example. The versions of Samba 4 currently published by the Samba project are expected to undergo more intensive changes in their further development than Samba 3. Univention has tested the required components for the provision of Active Directory-compatible domain services with Samba 4 and integrated them in UCS in close cooperation with the Samba team. Parallel to this, Samba 3 was integrated with Samba 4 for UCS. This means that the tried-and-tested file and print service of Samba 3 can be used at the same time as the use of Active Directory-compatible domain services.

An overview of the functional differences between Samba 3 and Samba 4 can be found in the Univention Wiki [wiki-samba4].

In addition to Samba 4, UCS also integrates an additional component for the synchronisation with Active Directory (AD): the Univention Active Directory Connector. In contrast to Samba 4, however, this operates a separate AD domain based on Microsoft Windows, which can be bidirectionally or unidirectionally synchronised, see Section 8.5.

8.2. Installation and components of a Samba domain

[Feedback](#) 

A Samba domain is composed of at least one domain controller. Windows clients and UCS member servers can join the trust context of the Samba domain as domain members. Such servers do not provide login services, but may offer file and print services, for example. Login to these servers is then performed against the UCS user data. UCS-based member servers are only provided via Samba 3.

Domain joining of Windows clients is described in Section 3.2.2.

Windows domain controllers can neither join the domain with Samba 3 nor with Samba 4. This function is planned for Samba 4 at a later point in time.

The name of the Samba domain is specified during installation of the master domain controller, depending on whether Samba is installed on the system or not. The name of the domain is saved in the Univention Configuration Registry variable `windows/domain`.

Samba 4 can not join an Active Directory Forest yet at this point. (For Samba 3 this function is not available because of the lack of support for the Active Directory technology.)

8.2.1. Installation and components of a Samba 3 domain

[Feedback](#) 

Samba 3 can be installed on all UCS domain controllers by selecting the installer component *NT-compatible domaincontroller (Samba 3)* or via subsequent installation of the *univention-samba* package. In the subsequent installation, the `univention-run-join-scripts` command must be run after installation.

On member servers, the installer component is called *Windows memberserver (Samba 3 / Samba 4)*. Alternatively, the packages **univention-samba** and **winbind** can be installed.

8.2.2. Installation and components of a Samba 4 domain

Samba 4 can only be installed on UCS domain controllers.

The *Active Directory-compatible domaincontroller (Samba 4)* component must be selected in the Univention Installer. Installation on the command line is performed with the ***univention-samba4*** package. On the system roles master domain controller and backup domain controller, the ***univention-s4-connector*** must also be installed (in installation via the Univention Installer, the connector package is automatically installed at the same time).

For the installation of a Samba member server, the **Windows Memberserver (Samba 3 / Samba 4)** component in the Univention Installer must be selected on a UCS member server or alternatively the packages ***univention-samba*** and ***winbind*** must be subsequently installed.

The signing of NTP replies should be set on all Samba domain controllers, see Section 7.4.11.

Samba 4 implements Active Directory directory services which allow a multi-master replication, i.e., the writing changes of several domain controllers are synchronised 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 or suspended.

8.3. Services of a Samba domain

8.3.1. Authentication services

In Samba 3, the user passwords are saved in the UCS LDAP. Users are authenticated against the LDAP directory when logging into the domain with their username and password, and can then access all the shared resources of the domain without having to enter their username and password again. Computers with any kind of Windows operating systems are authenticated in the same way as in Windows NT domains, via the NTLMv2 protocol.

Users who log on to a Windows system joined in a Samba 4 server 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. Synchronisation of the system times is essential for functioning Kerberos authentication.

User logins can only be performed on Microsoft Windows systems joined in the Samba domain. Domain joins are documented in Section 3.2.2.

Mixed environments containing both Samba 3 and Samba 4 are only supported for update scenarios and are described in more detail in the Univention Wiki [wiki-samba-update].

8.3.2. File services

The file services integrated in UCS support the provision of shares using the CIFS protocol (see Chapter 10). 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 4 requires a file system with XATTR support.

Samba 4 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 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.

8.3.3. 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 integration of the printer drivers is described in Chapter 11.

Samba 4 domain controllers can also provide print services. In this case, the restrictions described in Section 8.3.2 must be taken into consideration.

8.3.4. Univention S4 connector

[Feedback](#) 

In Samba 4, the Samba user accounts are managed completely by Samba. The synchronisation between the UCS LDAP and Samba occurs via an internal system service, the Univention S4 connector.

Further information on the status of the synchronisation can be found in the log file `/var/log/univention/connector-s4.log`.

8.3.5. NetBIOS name service

[Feedback](#) 

Windows systems use NetBIOS, a TCP/IP-based network protocol for resolving host names and for network communication. Samba provides NetBIOS functions with the `nmbd` system service.

NetBIOS computer names can have a maximum of 15 characters. The NetBIOS name of a UCS system corresponds to the host name by default. A different name can be configured with the Univention Configuration Registry variable `samba/netbios/name` and alias names defined with the Univention Configuration Registry variable `samba/netbios/aliases`.

In a native Active Directory environment, there are no NetBIOS services provided as standard. In an AD environment based on Samba 4, however, it is activated. This can be deactivated with the Univention Configuration Registry variable `samba4/service/nmb`.

Samba 3 and Samba 4 offer the same scope of functions in terms of NetBIOS services.

8.3.6. Name resolution using WINS

[Feedback](#) 

Similar to DNS in TCP/IP networks, the *Windows Internet Name Service (WINS)* is used for resolving NetBIOS names into IP addresses. In addition, WINS provides information on the services of the hosts.

WINS is used in Samba 3-based, NT-compatible domains; in Samba 4 domains, the name resolution generally occurs via DNS.

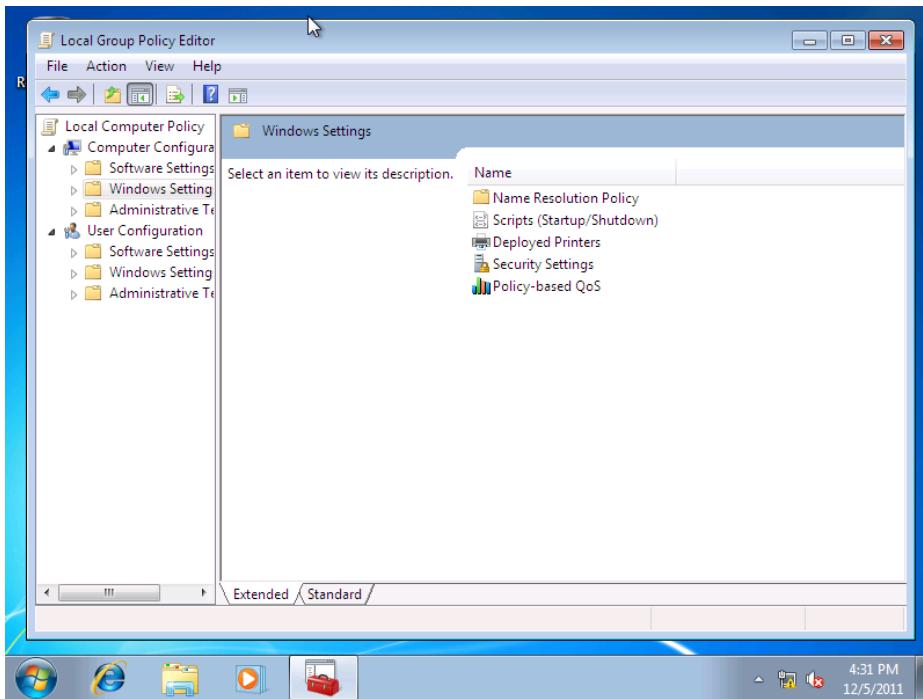
The WINS service can also be provided by Samba. WINS support is activated on the master domain controller in the default setting and can also be operated on another server by setting the Univention Configuration Registry variable `windows/wins-support`. WINS can only be operated without adjustments on one Samba server in the domain; distribution across several servers requires the setup of WINS replication. Information on the commissioning of the WINS replication can be found in the Univention Support database at <http://sdb.univention.de/1107>

The WINS server can assign Windows clients via a DHCP-NetBIOS policy, see Section 9.3.3.3.

8.4. Configuration and management of Windows desktops

8.4.1. Group policies

Figure 8.1. Group policy editor



Group policies are an Active Directory feature which allows the central configuration of computers and user settings. They are only supported by Samba 4.

The setting up of group policies can be performed with the *Remote Server Administration Tools (RSAT) for Windows 7*¹, or the *Remote Server Administration Tools (RSAT) for Windows 8*².

The group policies are saved in the SYSVOL share.

Windows Group Policies are not supported by Mac OS X clients.

8.4.2. Logon scripts / NETLOGON share

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 NETLOGON share must be available on all Samba domain controllers and always contain the same contents.

8.4.2.1. Samba 3

Under Samba 3, the directory `/var/lib/samba/netlogon` is set up as the Samba share *NETLOGON*.

¹<http://www.microsoft.com/downloads/details.aspx?FamilyID=7d2f6ad7-656b-4313-a005-4e344e43997d>

²<http://www.microsoft.com/de-de/download/details.aspx?id=28972>

Configuration of the file server for the home directory

In the default setting, all adjustments are made in the `/var/lib/samba/netlogon` directory on the master domain controller and synchronised hourly on all domain controllers with Samba installed via the `rsync` tool.

The Univention Configuration Registry variable `samba/logonscript` is available for defining a global logon script for all users. If this variable is set on a Samba server, then all users logging into this Samba server have the specified logon script assigned. The logon script can also be assigned user-specifically, see Section 5.1.

8.4.2.2. Samba 4

[Feedback](#) 

In Samba 4 the logon scripts are stored in `/var/lib/samba/sysvol/<Domainname>/scripts/` and similarly provided under the share name `NETLOGON`.

The `NETLOGON` share is replicated within the scope of the `SYSVOL` replication.

8.4.3. Configuration of the file server for the home directory

[Feedback](#) 

8.4.3.1. Configuration with Samba 3

[Feedback](#) 

As standard, the home directory of each user is shared by Samba and connected with the `I:` drive after login in Windows.

The Univention Configuration Registry variable `samba/homedirserver` can be used to specify the server on which the home directories should be stored; the Univention Configuration Registry variable `samba/homedirpath` can be used to specify the directory. These values will then be valid for all the users.

It is also possible to make individual assignment in the user settings - see Section 5.1 - with the setting **Windows home path**, e.g., `\ucs-file-server\smith`.

If instead of the user's UNIX home directory, a different UNIX directory is to be displayed as the home directory on the Windows drive, then this server and the home directory need to be entered in the **Windows home path** entry field.

8.4.3.2. Configuration with Samba 4

[Feedback](#) 

In Samba 4 the home directory can only be defined user-specifically in the Univention Management Console, see Section 5.1. This is performed with the setting **Windows home path**, e.g., `\ucs-file-server\smith`.

The multi edit mode of the Univention Management Console can be used to assign the home directory to multiple users at one time, see Section 4.2.3.3.

8.4.4. 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 synchronised onto the Samba server when the user logs off.

If the profile path is changed in the 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.

8.4.4.1. Samba 3

[Feedback](#) 

In Samba 3, the user profiles are saved in the `windows-profiles\<Windows-Version>` subdirectory on the Samba server that the user logged on to.

Univention Configuration Registry variable `samba/profileserver` can be used to specify another server and `samba/profilepath` to specify another directory. These settings must be set on all Samba domain controllers.

In the user management of the Univention Management Console, the input field **Windows profile directory** can be set to configure a different path or another server for the profile directory for the user.

Roaming profiles can be deactivated by configuring the Univention Configuration Registry variables `samba/profilepath` and `samba/profileserver` to *local* and restarting the Samba server.

8.4.4.2. Samba 4

[Feedback](#) 

No roaming profiles are used in the default setting in Samba 4.

Roaming profiles can be enabled per user; their use must be configured via the **Windows profile directory** field in the **Account** tab of the user management. The path entered may include Windows variables, e.g.,

```
sambaProfilePath: \\ucsmaster\%UserName%\windows-profiles
```

This path can also be set via a user template, see Section 5.6.

Alternatively, it is possible to configure the profile directory 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`.

8.4.5. Granting additional Windows privileges to users

[Feedback](#) 

The **Samba privileges** selection mask in the user management can be used to assign a user selected Windows system rights, such as the right to manage printers or to restart computers, see Section 5.1.

This feature is only available in Samba 3.

8.5. UCS Active Directory Connector

[Feedback](#) 

8.5.1. Introduction

[Feedback](#) 

The UCS Active Directory Connector (AD Connector for short) makes it possible to synchronise directory service objects between a Windows 2003/2008/2012 server under Active Directory (AD) and a Univention Corporate Server.

In the default setting, containers, organisational units, users, and groups are synchronised. Users have an exceptional position since the password cannot be queried via the LDAP protocol in Active Directory. A special service is installed on the Windows server for this purpose, which enables password synchronisation (see Section 8.5.2.3).

Information on the attributes configured in the basic setting and particularities to take into account can be found in the Section 8.5.4.

Computer accounts are not synchronised since Windows clients can only be a member in one domain.

Users can make access to services of both environments in a transparent way; this is due to the possibility of having the same user settings in both domains. After logging into a UCS domain, a 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 initialisation is performed after the first start of the connector. All the entries are read out of the UCS, converted in AD objects according to the mapping set and added (or modified if already present) on the UCS side. All objects are then read out of the AD and converted into UCS objects before being added/modified accordingly. The directory service servers are then queried further as long as no changes have been made. The AD connector can also be operated in a unidirectional mode.

Following the initial sync, additional changes are requested at a set interval. This interval is initially set to five seconds and can be adapted via the Univention Management Console configuration module.

If an object cannot be synchronised, it is "rejected". During start-up of the UCS AD connector, an attempt is made to write this object again. In addition, a further attempt is made to input the object again after several runs. The interval can be configured in the UMC configuration module, see Section 8.5.2.1. The presetting specifies ten cycles before the object is resynchronised. A restart of the UCS AD connector also triggers a resynchronisation of all previously rejected changes.

8.5.2. Setup of the UCS AD connector

[Feedback](#) 

The installation is performed installing the ***univention-ad-connector*** package.

The UCS AD connector can only be installed on a master domain controller or backup domain controller system.

Despite intensive tests it is not possible to rule out that the results of the synchronisation may affect the operation of a productive domain. The connector should therefore be tested for the respective requirements in a separate environment in advance.

All Active Directory and UCS servers in a connector environment must use the same time zone.

8.5.2.1. Basic configuration of the connector

[Feedback](#) 

The connector is configured using the wizard **UCS Active Directory Connector** of the 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.

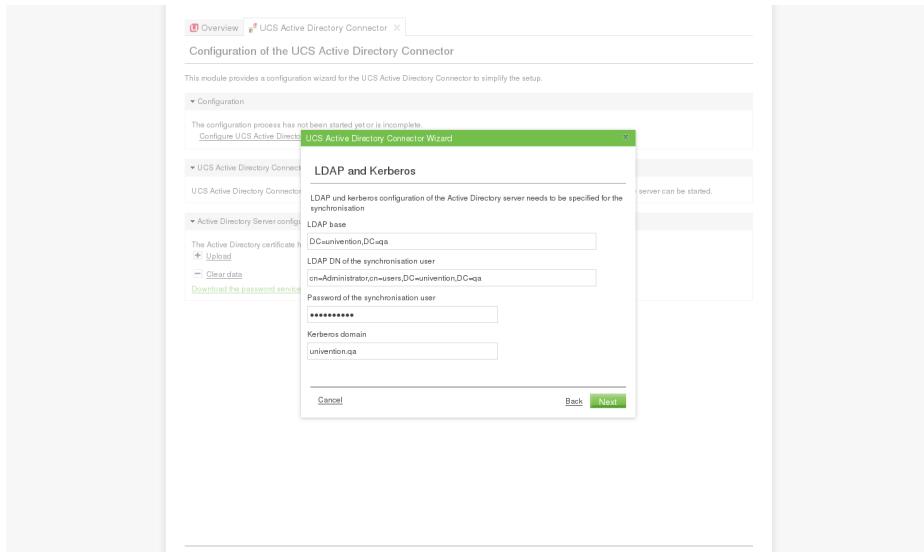
The connector's settings status is displayed under **Configuration**. The setup can be begun by clicking on **Configure UCS Active Directory Connector**.

The fully qualified host name of the Active Directory server must be given in the **Active Directory Server** field. If the hostname of the AD system is not resolvable for the UCS system, it is necessary to create a DNS host record for the AD system in the DNS management module of the Univention Management Console (see Section 9.2.2.3).

Alternatively, a static entry can also be adopted in `/etc/hosts` via Univention Configuration Registry, e.g.

```
ucr set hosts/static/192.168.0.100=w2k8-32.ad.example.com
```

Figure 8.2. Configuring the AD connector in UMC



If the **Automatic determination of the LDAP configuration** option is activated, settings such as the base DN of the Active Directory directory or the Kerberos domain of the AD system are queried automatically and do not need to be manually configured.

The LDAP DN of the user used for access to the Active Directory is configured in the **LDAP DN of synchronisation user** field. (The setting is saved in the Univention Configuration Registry variable `connector/ad/ldap/binddn`). When using the function for the automatic transfer of the base DN of the Active Directory, the Administrator account is entered by default for the base DN.

The password used for access is entered in the **Password of the synchronisation user** field and saved in a text file.

Now **Next** needs to be clicked. The presettings in the following dialogue are suitable for most environments and do not generally need to be changed.

Some group names are saved differently in Active Directory depending on the installation language of the server. The localisation used can be selected under **System language of Active Directory server**. Further information can be found in Section 8.5.4.2.

The AD connector can be operated in different modes, which can be selected in **Synchronisation mode**. Alongside bidirectional synchronisation, unidirectional replication towards the AD or UCS directory service is also possible.

Now **Next** needs to be clicked.

In **Poll Interval (seconds)** you can specify how long the system should wait after a run showing no changes before sending a new request.

Retry interval for rejected objects specifies after how many synchronisation intervals retained changes are imported subsequently.

The **Debug level of Active Directory connector** configures how much debugging information is written to the `/var/log/univention/connector.log` file. **Add debug output for functions** can be used to specify additionally whether further debug output is added for function calls.

Clicking on **Finish** adopts the configuration in Univention Configuration Registry. Changes are only adopted following a restart of the UCS AD connector, see Section 8.5.2.4.

8.5.2.2. Importing the SSL certificate of the Active Directory

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. Two 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 concern the communication with the password service (see Section 8.5.2.3); this is always encrypted.

8.5.2.2.1. Exporting the certificate on Windows 2003

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**.

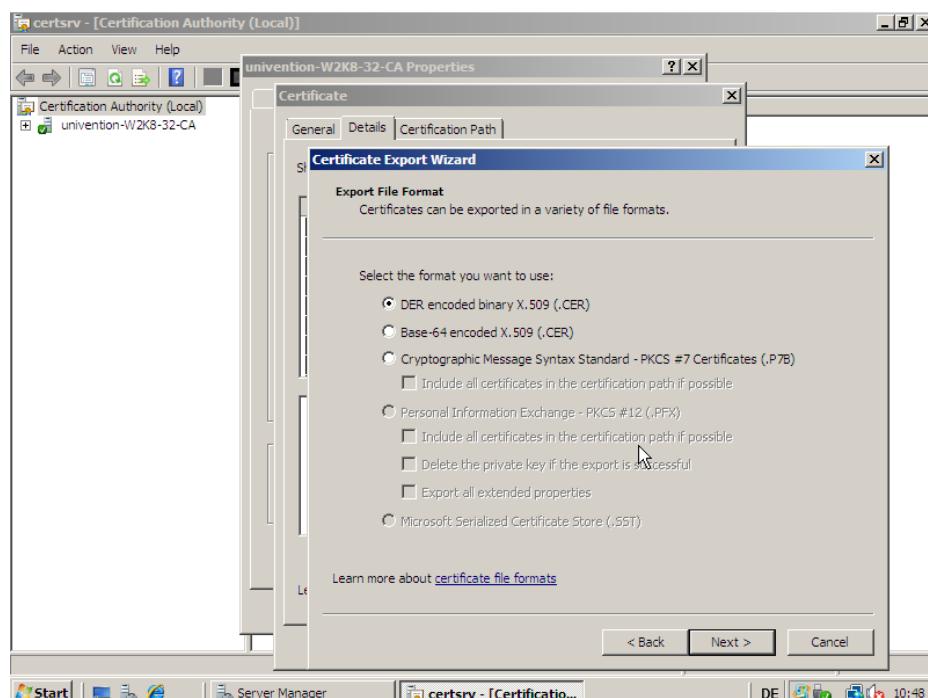
The AD server should be rebooted after the installation.

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**.

8.5.2.2.2. Exporting the certificate on Windows 2008

If the certificate service is not installed, this must be installed before proceeding.

Figure 8.3. 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 -> Right click on the name of the generated certificate -> Properties -> View Certificate -> Details -> Copy to File... -> Next -> DER encoded binary X.509 (.CER) -> Next Select an arbitrary file name and save path -> Next -> Finish**

8.5.2.2.3. Copying the Active Directory certificate to the UCS system

[Feedback](#)

The SSL AD certificate should now be imported into the UCS system using the Univention Management Console wizard.

This is done by clicking on **Upload** in the sub menu **Active Directory Server configuration**.

This opens a window in which a file can be selected, which is being uploaded and integrated into the AD connector.

8.5.2.3. Setting up the password service on the AD system

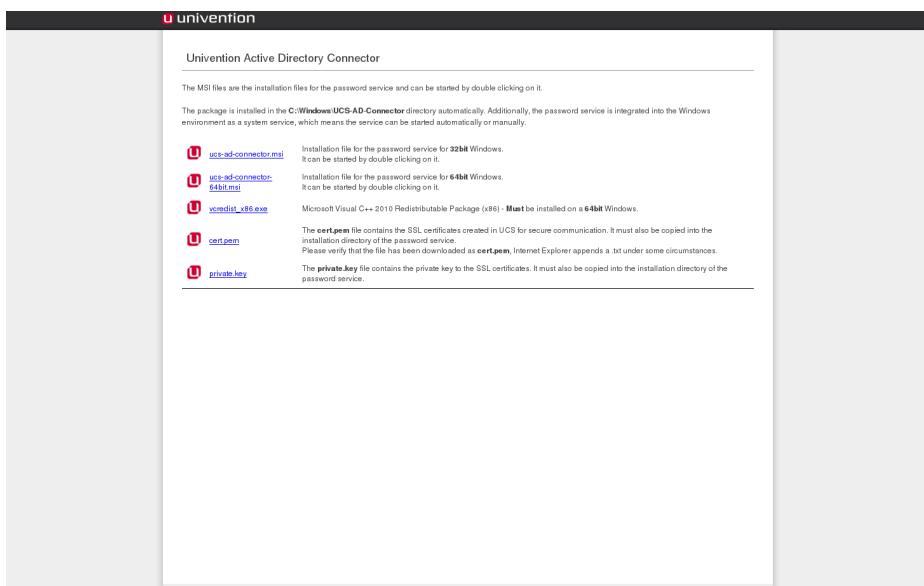
[Feedback](#)

Active Directory prohibits the request of passwords via the LDAP protocol, which requires the installation of a package on the Windows server.

The installation packages are also provided via the setup wizard Univention Management Console.

Selecting **Download the password service for Windows and the UCS certificate** opens a new browser window in which five files are available to download:

Figure 8.4. Packages for the password service



- `ucs-ad-connector.msi` (for 32bit Windows)
- `ucs-ad-connector-64bit.msi` (for 64bit Windows)
- Microsoft Visual C++ 2010 Redistributable Package (x86)

- `private.key`
- `cert.pem`

On 64-bit Windows versions, the *Microsoft Visual C++ 2010 Redistributable Package (x86)* must be installed before the installation of the AD connector.

The MSI files are the installation files for the password service and can be started by double clicking on it.

The package is installed in the `C:\Windows\UCS-AD-Connector` directory automatically. Additionally, the password service is integrated into the Windows environment as a system service, which means the service can be started automatically or manually. After the installation the password service is configured for automatic startup.

The `private.key` and `cert.pem` files contain the SSL certificates created in UCS for secure communication with the password service. They must also be copied into the installation directory of the password service. The password service must then be restarted.

During a standard installation in Windows 2008 the Windows firewall blocks the access to the AD connector. This must either be deactivated in **System settings** or Port 6670/TCP authorised.

8.5.2.4. Starting/Stopping the Active Directory connector

[Feedback](#) 

The connector can be started using **Start Active Directory connector** and stopped using **Stop Active Directory connector**. Alternatively, the starting/stopping can also be performed with the `/etc/init.d/univention-ad-connector` init script.

8.5.2.5. 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.

8.5.3. Additional tools / Debugging connector problems

[Feedback](#) 

The AD connector provides the following tools and log files for diagnosis:

8.5.3.1. 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
```

8.5.3.2. univention-connector-list-rejected

[Feedback](#) 

This tool lists the DNs of non-synchronised 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 synchronised by AD.

This script may display an error message or an incomplete output if the AD connector is in operation.

8.5.3.3. Logfiles

[Feedback](#) 

For troubleshooting when experiencing synchronisation 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
```

The status notifications of the password service on the AD side are logged in the C:\\Windows\\\\UCS-AD-Connector\\\\UCS-AD-Connector.log file.

8.5.4. Details on preconfigured synchronisation

[Feedback](#) 

All containers which are ignored due to corresponding filters are exempted from synchronisation as standard. This can be found in the /etc/univention/connector/ad/mapping configuration file under the *global_ignore_subtree* setting.

8.5.4.1. Containers and organisational units

[Feedback](#) 

Containers and organisational units are synchronised 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**).

8.5.4.1.1. Particularities

[Feedback](#) 

- Containers or organisational units deleted in AD are deleted recursively in UCS, which means that any non-synchronised subordinate objects, which are not visible in AD, are also deleted.

8.5.4.2. Groups

[Feedback](#) 

Groups are synchronised 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 synchronised.

8.5.4.2.1. 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 synchronisation of the SID or the RID.
- Groups which were configured as *Local Group* under **Samba group type** in Univention Directory Manager are synchronised 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 initialisation, the members are synchronised, but not the position in LDAP. If one of these groups is migrated on one side, the target container

Details on preconfigured synchronisation

on the other side is identical, so that the DNs of the groups can no longer be differentiated from this point onwards.

- Certain group names are converted using a mapping table so that, for example in a German language set-up, the UCS group *Domain Users* is synchronised 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 synchronised 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.

8.5.4.3. Users

[Feedback](#)

Users are synchronised 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), *Organisation*, *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 synchronised. *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.

8.5.4.3.1. Particularities

[Feedback](#)

- Users are also identified using the name, so that for users created before the first synchronisation on both sides, the same process applies as for groups as regards the position in LDAP.
- The synchronisation of the password expiry date and the *Change password on next login* user option occurs on the UCS side on the Samba level alone. If a password change is initiated by Univention Directory Manager, but the password changed in Active Directory, the expiration details for the Kerberos and Posix passwords are not changed, so that the user must change his password again if he, for example, logs on to a thin client.
- 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 synchronisation from UCS to AD the users are therefore always a member in this group.

8.6. Trust relationships

Trust relationships between domains make it possible for users from one domain to log on to computers from another domain.

Trust settings are only supported by Samba 3.

If a Windows domain trusts a Samba 3 domain, there is also the possibility to log on to the Samba domain alongside the Windows domain when logging on to computers belonging to the Windows domain.

If a Samba 3 domain trusts a Windows domain, users from the Windows domain enter the user name <name-of-windows-domain>+<username> when logging on to a Linux computer belonging to the Samba domain.

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 using broadcasts or WINS.

8.6.1. Windows domain trusts Samba 3 domain

A *Domain Trust Account* with a name reflecting the NetBIOS name of the Windows domain and a password issued for the account must be created in the computer management module of Univention Management Console. The password quality requirements which may apply to Windows domains must be observed.

Trust settings can only be set up on domain controllers.

A trust relationship must be created on the Windows PDC.

The trust relationship between the Windows domain and the Samba domain can be removed by deleting the trust relationship on the Windows PDC and the domain trust account in the Univention Management Console.

8.6.2. Samba 3 domain trusts Windows domain

The following steps are used to set up the trust setting on a slave domain controller as a *root* user:

The **winbind** package must be installed. Winbind maps UNIX IDs to Windows users and groups

A trust relationship must be created on the Windows PDC.

If Univention Firewall is used, replies to NetBIOS broadcasts need to be allowed:

```
echo "iptables -I INPUT 1 -p udp --sport 137 -j ACCEPT" \
    >> /etc/security/packetfilter.d/50_local.sh
/etc/init.d/univention-firewall restart
```

The instructions available on UCS Bug 25254 should currently be considered.

The trust relationship is now initiated and Winbind restarted. This command must be run on all Samba login servers:

```
net rpc trustdom establish <Windows domain>
/etc/init.d/winbind restart
```

The following command can be used to check that the trust relationship has been added correctly:

```
net rpc trustdom list
```


Chapter 9. IP and network management

9.1. Network objects	134
9.2. Administration of DNS data with Bind	135
9.2.1. Configuration of the Bind nameserver	136
9.2.1.1. Configuration of Bind debug output	136
9.2.1.2. Configuration of the data backend	136
9.2.1.3. Configuration of zone transfers	136
9.2.2. Administration of DNS data in the Univention Management Console	137
9.2.2.1. Forward lookup zone	137
9.2.2.2. CNAME-Record (Alias-Records)	139
9.2.2.3. A/AAAA records (host records)	139
9.2.2.4. Service records	139
9.2.2.5. Reverse lookup zone	141
9.2.2.6. Pointer record	141
9.3. IP assignment via DHCP	142
9.3.1. Introduction	142
9.3.2. Composition of the DHCP configuration via DHCP LDAP objects	143
9.3.2.1. Administration of DHCP services	143
9.3.2.2. Administration of DHCP server entries	143
9.3.2.3. Administration of DHCP subnets	143
9.3.2.4. Administration of DHCP pools	144
9.3.2.5. Registration of computers with DHCP computer objects	145
9.3.2.6. Management of DHCP shared networks / DHCP shared subnets	145
9.3.3. Configuration of clients via DHCP policies	146
9.3.3.1. Setting the gateway	146
9.3.3.2. Setting the DNS servers	146
9.3.3.3. Setting the WINS server	146
9.3.3.4. Configuration of the DHCP lease	147
9.3.3.5. Configuration of boot server/PXE settings	147
9.3.3.6. Further DHCP policies	148
9.4. Packet filter with Univention Firewall	148
9.5. Web proxy for caching and policy management / virus scan	148
9.5.1. Installation	149
9.5.2. Caching of web content	149
9.5.3. Logging proxy accesses	149
9.5.4. Restriction of access to permitted networks	149
9.5.5. Configuration of the ports used	150
9.5.5.1. Access port	150
9.5.5.2. Permitted ports	150
9.5.6. User authentication on the proxy	150
9.5.7. Filtering/policy enforcement of web content with Dansguardian	151
9.5.8. Definition of content filters for Dansguardian	152

IP addresses for the computer systems managed in a UCS domain can be centrally managed via the Univention Management Console and assigned via DHCP.

With *network objects* one can manage the features of a network centrally, e.g., the available IP addresses and the DNS and DHCP zones in which a system is located.

The assignment of IP addresses via DHCP and the DNS resolution are also integrated in UCS. The data used for this, e.g., the MAC address, are stored in the LDAP directory.

Incoming and outgoing network traffic can be restricted via the *Univention Firewall* based on IPTables.

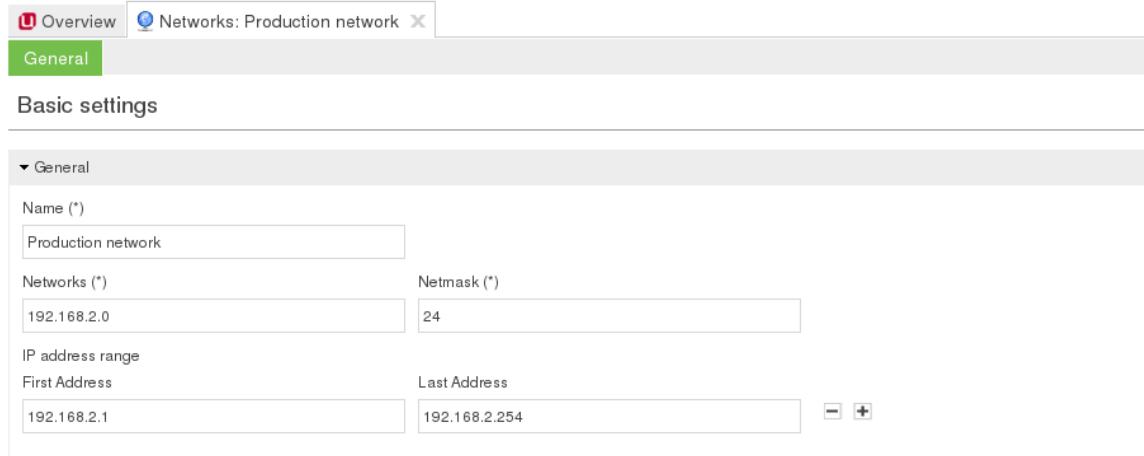
The integration of the proxy server Squid allows the caching of web contents and the enforcement of content policies for web access.

9.1. 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 9.1. Creating a network object



The screenshot shows the 'General' tab of the 'Networks: Production network' configuration page. The 'Name (*)' field is set to 'Production network'. The 'Networks (*)' field contains '192.168.2.0' and the 'Netmask (*)' field contains '24'. In the 'IP address range' section, the 'First Address' is '192.168.2.1' and the 'Last Address' is '192.168.2.254'. There are minus and plus buttons for adjusting the range.

For example, it is possible to define a network object *Workstation network* which encompasses the IP addresses from 192.168.2.0 to 192.168.2.254. 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 *Networks* module of the Univention Management Console.

General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

Table 9.1. 'General' tab

Attribut	Beschreibung
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., 192.168.1.0.
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.
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. When no IP range is entered here, the system automatically uses the range given by the network and the subnet mark entered.

Attribut	Beschreibung
	<p>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.</p> <p>The zones are also administrated in the Univention Management Console, see Section 9.2.2.1.</p> <p>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.</p>
DNS forward lookup zone	<p>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.</p>
DNS reverse lookup zone	<p>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.</p> <p>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.</p> <p>The DHCP service settings are also administrated in the Univention Management Console, see Section 9.3.2.</p> <p>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.</p>

9.2. 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 7.2.4.

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 synchronise 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 amongst 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 synchronise 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.

9.2.1. Configuration of the Bind nameserver

[Feedback](#) 

9.2.1.1. 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 9.2.1.2) 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-level].

9.2.1.2. 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 the 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.
- Samba 4 provides an Active Directory domain. Active Directory is closely connected with DNS, for DNS updates of Windows clients or the localisation 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 the Univention Management Console in both cases.

9.2.1.3. 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*.

9.2.2. Administration of DNS data in the Univention Management Console

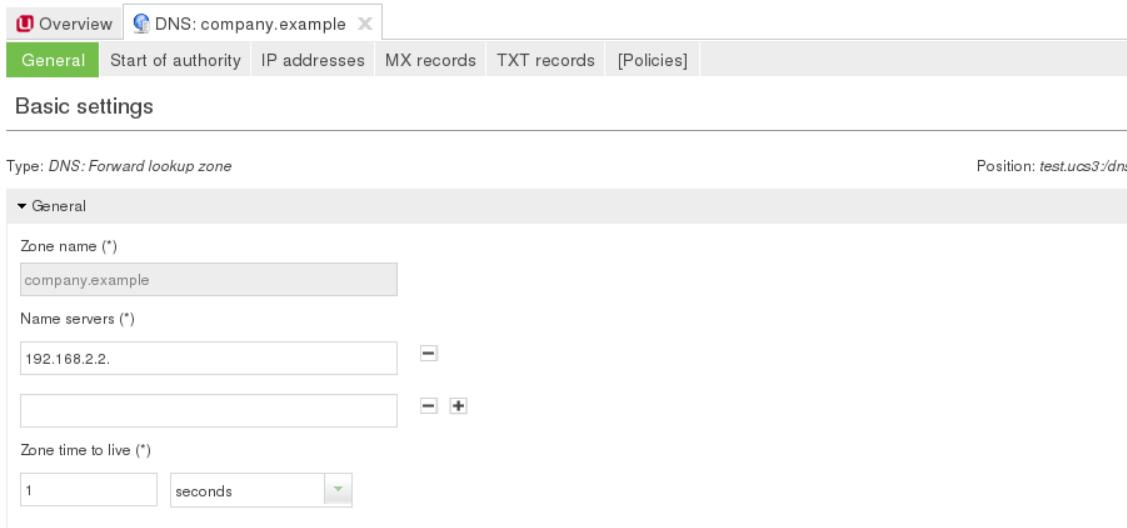
DNS files are stored in the *cn=dns,<base-dn>* container as standard. Forward and reverse lookup zones are stored directly in the container. Additional DNS objects such as pointer records can be stored in the respective zones.

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.

9.2.2.1. Forward lookup zone

Forward lookup zones contain information which is used to resolve DNS names into IP addresses. They are managed in the *DNS* module of the Univention Management Console. To create an additional forward lookup zone, **None** must be entered as **Superordinate**.

Figure 9.2. Configuring a forward lookup zone in UMC



General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

Table 9.2. 'General' tab

Attribut	Beschreibung
Zone name	This is the complete name of the DNS domain for which the zone will be responsible. 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 9.3. 'Start of authority' tab

Attribut	Beschreibung
Contact person	The e-mail address of the person responsible for administrating the zone.
Serial number	<p>Other DNS servers use the serial number to recognise 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.</p> <p>There are two commonly used patterns for this serial number:</p> <ul style="list-style-type: none"> • Start with 1 and increment the serial number with each change • 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. <p>If the serial number is not changed manually, it will be increased automatically with every change.</p>
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	<p>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.</p> <p>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.</p>
Minimum time to live	The minimum time to live specifies in seconds how long other servers can cache no-such-domain (NXDOMAIN) answers. This value cannot be set at more than 3 hours, the preset value is 3 hours.

Table 9.4. Reiter 'IP addresses'

Attribut	Beschreibung
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 9.5. 'MX records' tab

Attribut	Beschreibung
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.

Attribut	Beschreibung
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 9.6. 'TXT records' tab

Attribut	Beschreibung
TXT record	Descriptive text for this zone. Text records must not contain umlauts or other special characters.

9.2.2.2. CNAME-Record (Alias-Records)

[Feedback](#)

CNAME records / alias records are managed in the *DNS* module of the Univention Management Console. To create an additional record, a forward lookup zone must be entered as the **Superordinate** and the **Alias record** object type selected.

General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

Table 9.7. 'General' tab

Attribut	Beschreibung
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.

9.2.2.3. A/AAAA records (host records)

[Feedback](#)

Host records are managed in the *DNS* module of the Univention Management Console. To create an additional record, a forward lookup zone must be entered as the **Superordinate** and the **Host record** object type selected.

General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

When adding or editing a computer object a host record can be created automatically or edited.

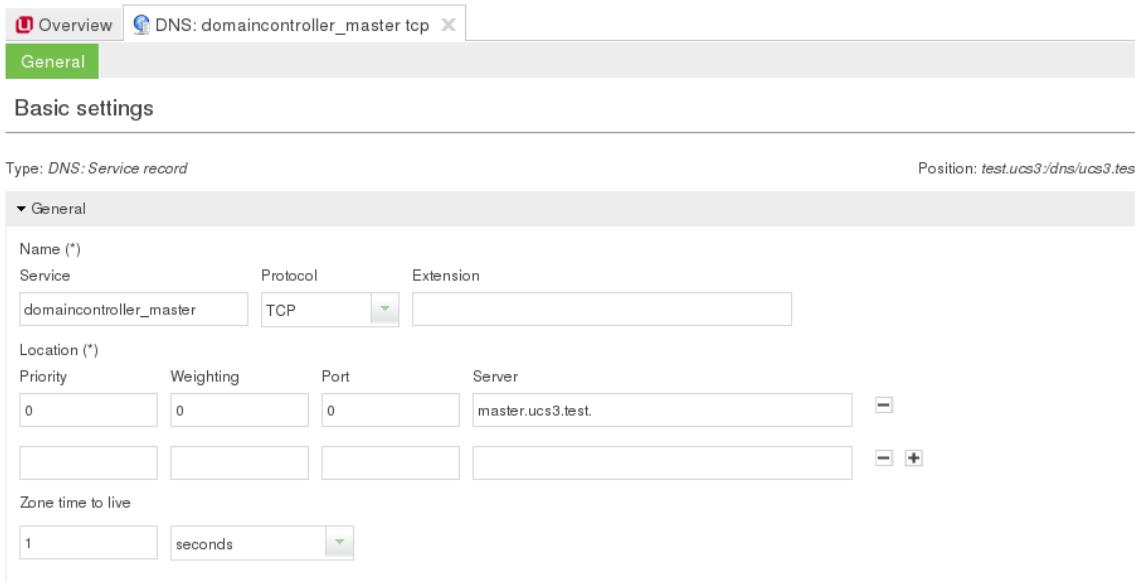
Table 9.8. 'General' tab

Attribut	Beschreibung
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.
Zone time to live	The time to live specifies in seconds how long these files may be cached by other DNS servers.

9.2.2.4. Service records

[Feedback](#)

Service records are managed in the *DNS* module of the Univention Management Console. To create an additional record, a forward lookup zone must be entered as the **Superordinate** and the **Service record** object type selected.

Figure 9.3. Configuring a service records


Type: *DNS: Service record* Position: *test.ucs3/dns/ucs3.test*

General

Name (*)

Service	Protocol	Extension
domaincontroller_master	TCP	

Location (*)

Priority	Weighting	Port	Server
0	0	0	master.ucs3.test.

Zone time to live

1	seconds
---	---------

General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

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 9.9. 'General' tab

Attribut	Beschreibung
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. Example: <i>Server1</i> has a priority of 1 and a weight function of 1, whilst <i>Server2</i> also has a priority of 1, but has a weight function of 3. In this case, <i>Server2</i> will be used three times as often as <i>Server1</i> . 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.

Attribut	Beschreibung
	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.

9.2.2.5. Reverse lookup zone

[Feedback](#)

A reverse lookup zone is used to resolve IP address into host names. They are managed in the *DNS* module of the Univention Management Console. To create an additional reverse lookup zone, **None** must be entered as **Superordinate**.

Table 9.10. 'General' tab

Attribut	Beschreibung
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 <i>192.168.1.0</i> to <i>192.168.1.255</i> , <i>192.168.1</i> 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 synchronise themselves with the authoritative server via zone transfers. The entry which defines such a zone is called a SOA record in DNS terminology.

Table 9.11. 'Start of authority' tab

Attribut	Beschreibung
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 9.2.2.1.
Refresh interval	See the documentation on forward lookup zones in Section 9.2.2.1.
Retry interval	See the documentation on forward lookup zones in Section 9.2.2.1.
Expiry interval	See the documentation on forward lookup zones in Section 9.2.2.1.
Minimum time to live	See the documentation on forward lookup zones in Section 9.2.2.1.

9.2.2.6. Pointer record

[Feedback](#)

Pointer records are managed in the *DNS* module of the Univention Management Console. To create an additional record, a forward lookup zone must be entered as the **Superordinate** and the **Pointer** object type selected.

General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

Table 9.12. 'General' tab

Attribut	Beschreibung
Address	The last octet of the computer's IP address (depends on network prefix, see example below).

Attribut	Beschreibung
Pointer	<p>The computer's fully qualified domain name with a full stop at the end.</p> <p>In a network with a 24-bit network prefix (subnet mask 255.255.255.0) a pointer should be created for the <i>client001</i> computer with the IP address 192.168.1.101. 101 must then be entered in the Address field and <i>client001.company.com.</i> in Pointer.</p> <p>Example:</p> <p>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). <i>client001.company.com.</i> also needs to be entered in the Pointerfield here.</p>

9.3. IP assignment via DHCP

[Feedback](#)

9.3.1. 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.

Every UCS server on which the *DHCP server* component was selected during installation or the **univention-dhcp** package has been subsequently installed 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 variable 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, instead they are created 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 variable 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 in another IP range. DHCP pools can only be created below DHCP subnet objects.

If several subnets use the same common, physical 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.

9.3.2. Composition of the DHCP configuration via DHCP LDAP objects

[Feedback](#) 

9.3.2.1. Administration of DHCP services

[Feedback](#) 

DHCP services are managed in the *DHCP* module of the Univention Management Console. To create an additional service, **None** must be entered as **Superordinate**.

General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

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 9.3.3.

Table 9.13. 'General' tab

Attribut	Beschreibung
Service name	An unambiguous name for the DHCP service can be entered in this input field, e.g., <i>company.example</i> .

9.3.2.2. Administration of DHCP server entries

[Feedback](#) 

DHCP servers are managed in the *DHCP* module of the Univention Management Console. To create an additional DHCP server entry, a DHCP service must be selected as **Superordinate**.

General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

Each server which should offer the DCHP 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.

Table 9.14. 'General' tab

Attribut	Beschreibung
Server name	The computer name that the DHCP service should offer is entered in this input field, e.g., <i>ucs-master</i> . A server cannot be entered in more than one DHCP service at the same time.

9.3.2.3. Administration of DHCP subnets

[Feedback](#) 

DHCP subnets are managed in the *DHCP* module of the Univention Management Console. To create an additional DHCP server entry, a DHCP service must be selected as **Superordinate**.

General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

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 9.3.2.6).

Table 9.15. 'General' tab

Attribut	Beschreibung
Subnet address	The IP address of the subnet must be entered in dot-decimal form in this input field, e.g., <i>192.168.1.0</i> .
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. Caution 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 **Router** input field on the **Policy: DHCP Routing** tab (unless other entries are performed at lower levels).

9.3.2.4. Administration of DHCP pools

[Feedback](#)

DHCP pools can only be managed via the *LDAP directory* module of the Univention Management Console. 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 LDAP object**.

If DHCP pools are created in a subnet, no IP address range should be defined in the subnet entry. These should only be created in the pool entries.

Table 9.16. 'General' tab

Attribut	Beschreibung
Name	An unambiguous name for the DHCP pool can be entered in this input field, e.g., <i>testnet.compaby.example</i> .
Dynamic range	Here you can enter the IP addresses in dot-decimal form that are to be dynamically assigned.

Table 9.17. 'Advanced settings' tab

Attribut	Beschreibung
	Configuration options can be set in this tab which are only needed for a few scenarios.

9.3.2.5. 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. Only known computers receive fixed IP addresses from the DHCP service; unknown computers only receive 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 *DHCP* module of the Univention Management Console. To create an additional DHCP server entry, a DHCP service must be selected as **Superordinate**.

General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

Table 9.18. 'General' tab

Attribut	Beschreibung
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.
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 DNS server.

9.3.2.6. 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 *DHCP* module of the Univention Management Console. To create an additional DHCP server entry, a DHCP service must be selected as **Superordinate**.

General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

A shared subnet (see below) should be entered in the shared network under all circumstances as otherwise the DHCP service will need to be ended and restarted if there is an empty shared network in its configuration.

Table 9.19. 'General' tab

Attribut	Beschreibung
Shared network name	A name for the shared network should 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 should 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 *LDAP directory* module of the Univention Management Console. To do so, one must always be in a DHCP subnet 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 LDAP object**.

9.3.3. Configuration of clients via DHCP policies

[Feedback](#) 

9.3.3.1. Setting the gateway

[Feedback](#) 

The default gateway can be specified via DHCP with a *DHCP routing* policy, which is managed in the **Policies** module of the Univention Management Console.

General information on policy management can be found in Section 4.5.

Table 9.20. 'General' tab

Attribut	Beschreibung
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.

9.3.3.2. 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 **Policies** module of the Univention Management Console.

General information on policy management can be found in Section 4.5.

Table 9.21. 'General' tab

Attribut	Beschreibung
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.

9.3.3.3. 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 **Policies** module of the Univention Management Console.

General information on policy management can be found in Section 4.5.

Table 9.22. 'General' tab

Attribut	Beschreibung
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 RFC1001 and RFC1002. Attention must be paid to uppercase and lowercase when entering the NetBIOS scope.
NetBIOS node type	This field sets the node type of the client. Possible values are: <ul style="list-style-type: none"> • 1 B-node (Broadcast: no WINS) • 2 P-node (Peer: only WINS) • 4 M-node (Mixed: first Broadcast, then WINS) • 8 H-node (Hybrid: first WINS, then Broadcast)

[Feedback](#)

9.3.3.4. Configuration of the DHCP lease

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 **Policies** module of the Univention Management Console.

General information on policy management can be found in Section 4.5.

Table 9.23. 'General' tab

Attribut	Beschreibung
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.

[Feedback](#)

9.3.3.5. Configuration of boot server/PXE settings

A *DHCP Boot* policy is used to assign computers configuration parameters for booting via BOOTP/PXE. They are managed in the **Policies** module of the Univention Management Console.

General information on policy management can be found in Section 4.5.

Table 9.24. 'Boot' tab

Attribut	Beschreibung
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

Attribut	Beschreibung
	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/).

9.3.3.6. 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 UCS-based DNS service, but only with external servers.
- *DHCP Allow/Deny* allows the detailed assignment of access rights on the DHCP servers.
- *DHCP statements* allows the configuration of different options, which are only required in exceptional cases.

9.4. 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. 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 to *true*

9.5. 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 proxiss and the integration of a virus scan engine - are documented in the extended network management documentation [ext-doc-net].

9.5.1. Installation

[Feedback](#) 

The web proxy can be installed on any UCS server system role with the package ***univention-squid***.

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 9.5.5.1); port 3128 is set as default.

If changes are made to the configuration, Squid must be restarted. This can be performed either via the Univention Management Console or the command line:

```
/etc/init.d/squid3 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/squid3/local.conf`.

Dansguardian can be installed via the package ***univention-dansguardian***, see Section 9.5.7.

9.5.2. 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 by setting this to *no*. Squid must then be restarted.

9.5.3. Logging proxy accesses

[Feedback](#) 

All accesses performed via the proxy server are stored in the logfile `/var/log/squid3/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`.

9.5.4. 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 192.168.1.10 and the network mask 255.255.255.0 is available on the computer on which Squid is installed, only computers from the network 192.168.1.0/24 can access the proxy server. Additional networks can be specified via the Univention Configuration Registry variable. When doing so, the CIDR notation must be used; several networks should be separated by blank spaces.

Example:

```
univention-config-registry set squid/allowfrom="192.168.2.0/24  
192.168.3.0/24"
```

Once Squid has been restarted, access is now permitted from the networks 192.168.2.0/24 and 192.168.3.0/24. If configured to *all*, proxy access is granted from all networks.

Configuration of the ports used

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.

9.5.5. Configuration of the ports used

[Feedback](#) 

9.5.5.1. 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. When the Univention firewall is used, the packet filter configuration must also be adjusted.

When using the content and virus scanner (see Section 9.5.7) 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.

9.5.5.2. 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).

9.5.6. 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 unauthorised 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 authentication must firstly be enabled. There are a number of possible mechanisms for this: It can either be performed directly against the LDAP server or an NTLM authentication can be performed. The latter offers the advantage that it is not necessary to re-enter the password when Windows computers attempt to access the proxy.

To activate the authentication via LDAP, the Univention Configuration Registry variable `squid/basicauth` and for NTLM the Univention Configuration Registry variable `squid/ntlmauth` must be set to `yes`.

As standard, an NTLM authentication is performed for every HTTP query. If for example the website `http://www.univention.de` is opened, the subpages and images are loaded in addition to the actual HTML page. The NTML 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 HMTL queries in the same domain. This is enabled by default. To avoid the display of multiple authentication dialogues on Windows clients, which haven't joined the UCS domain yet, the feature can be disabled with `false`.

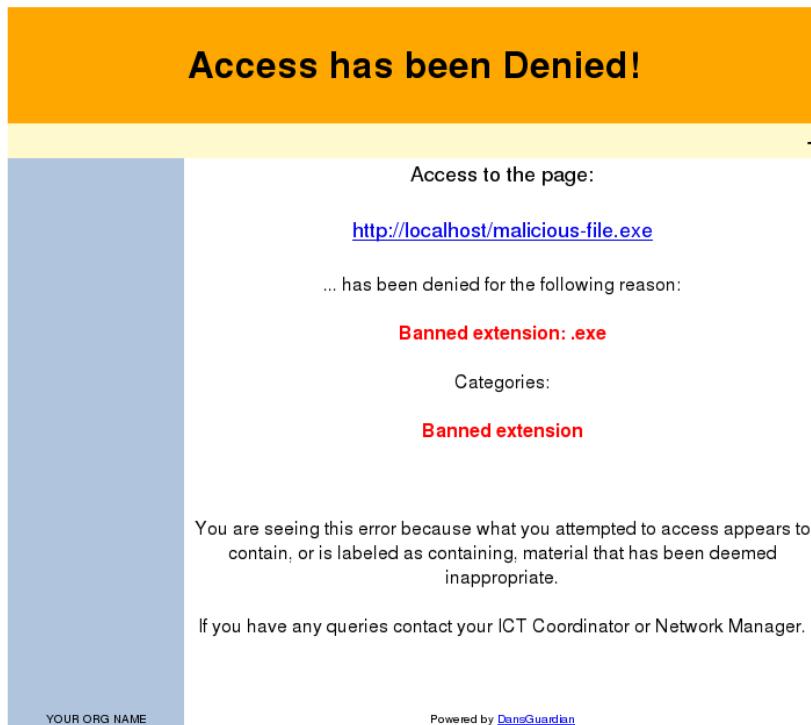
In the standard setting all users can access the proxy. The Univention Configuration Registry variable `squid/auth/allowed_groups` can be used to limit the proxy access to one or several groups. If several groups are specified, they must be separated by a semicolon.

9.5.7. Filtering/policy enforcement of web content with Dansguardian

[Feedback](#)

Dansguardian accepts website requests from the network and checks whether access by the request's sender 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 9.4. 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.

9.5.8. 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 realise 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 **System services** module of the Univention Management Console or on the commandline 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 9.25. UCR variables for filter rules

UCR variable	Description
<code>dansguardian/groups/<group>/banned/extensions</code>	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: <code>'.doc .xls .exe'</code> .
<code>dansguardian/groups/<group>/banned/mimetypes</code>	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: <code>audio/mpeg application/zip</code>
<code>dansguardian/groups/<group>/banned/sites</code>	This can be used to block complete web sites. Example: <code>illegal-example-website.com</code>
<code>dansguardian/<group>/banned/urls</code>	In contrast to the previous parameter, this can be used to block only specific URLs of websites.
<code>dansguardian/<group>/exception/urls</code>	The access to the URLs specified here is not filtered by Dansguardian.
<code>dansguardian/<group>/exception/sites</code>	The access to the web sites specified here is not filtered by Dansguardian.
<code>dansguardian/banned-ipaddresses</code>	This variable makes it possible to exclude individual clients (based on the IP address) from accessing the proxy server
<code>dansguardian/exception-ipaddresses</code>	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

UCR variable	Description
	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!

Chapter 10. Fileshare management

10.1. Access rights to data in shares	155
10.2. Management of shares in the Univention Management Console	156
10.3. Support for MSDFS	162

UCS supports the central management of directory shares. A share registered in the 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 the Univention Management Console are currently only provided via NFSv3.

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`.

10.1. 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.

Access control lists allow even more complex permission models. The configuration of ACLs is described in the Univention SDB [sdb-acls].

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 10.2).

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.

Changes to the permissions of a shared directory performed directly in the file system are not forwarded to the LDAP directory. If the share object is edited within the 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.

10.2. Management of shares in the Univention Management Console

File shares are managed in the *Shares* module of the Univention Management Console.

General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

When adding/editing/deleting a share, it is entered, modified or removed in the `/etc/exports` file and/or the Samba configuration.

Figure 10.1. Creating a share in UMC

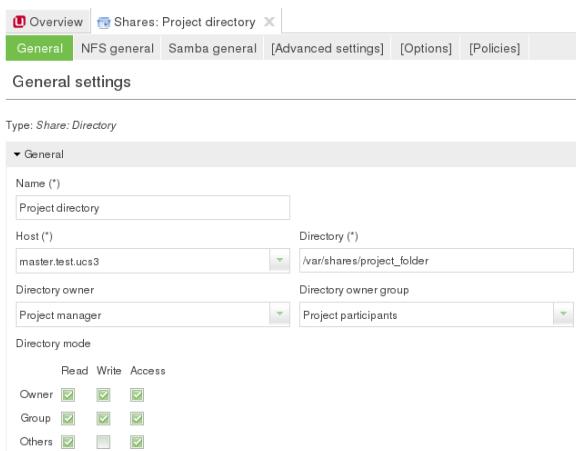


Table 10.1. 'General' tab

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. This setting cannot be subsequently edited. 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. This setting cannot be subsequently edited.
Directory	<p>The absolute path of the directory to be shared, without quotation marks (this also applies if the name includes special characters such as spaces). If the directory does not exist, it will be created automatically on the selected server.</p> <p>If the Univention Configuration Registry variable <code>listener/shares/rename</code> is set to <code>yes</code>, the contents of the existing directory are moved when the path is modified.</p> <p>No shares can be created in and below <code>/proc</code>, <code>/tmp</code>, <code>/root</code>, <code>/dev</code> and <code>/sys</code> and no files can be moved there.</p>
Directory owner	The user to whom the root directory of the share should belong, see Section 10.1.

Attribute	Description
Directory owner group	The group to whom the root directory of the share should belong, see Section 10.1.
Directory mode	The read, write and access permissions for the root directory of the share, see Section 10.1.

Table 10.2. 'NFS General' 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.
Redirect root access	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 <i>nobody</i> .
Caution	
	The local group <i>staff</i> , which is by default empty, owns privileges which come quite close to <i>root</i> 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 synchronisation	The synchronisation mode for the share. The <i>sync</i> setting is used to write data directly on the underlying storage device. The opposite setting - <i>async</i> - can improve performance but also imparts the risk of data loss if the server is shut down incorrectly.
Allowed hosts	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 10.3. 'Samba General' 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 adding a directory share, Univention Management Console adopts the name entered in the Name field of the General tab as the default. The name must be composed of letters, numerals, full stops or blank spaces and must begin and end with a letter or numeral.
Browsable	Specifies whether the share in question is to show up on Windows clients within the network environment.
Public	Permits access to this share without a password. Every access is carried out by means of the common guest user <i>nobody</i> .

Attribute	Description
MSDFS Root	This option is documented in Section 10.3.
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 10.1.
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.
Postexec script	A script or command which is to be executed on the server if the connection to this share is finished.
Preeexec script	A script or command which is to be executed on the server each time a connection to this share is established.

Table 10.4. 'Samba permissions' tab (advanced settings)

Attribute	Description
Samba write access	Permits write access to this share.
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. 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 authorised 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 accesses to a share at file server level beyond the file permissions. 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: <ul style="list-style-type: none">• 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
	<ul style="list-style-type: none"> • 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. • 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 unauthorised users, access is denied.
Allowed hosts	Names of computers which are authorised 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., 192.168.0.0/255.255.255.0.
Denied hosts	The opposite to the authorised computers. If a computer appears in both lists, the computer is permitted to access the Samba share.
Hide files	<p>Files and directories to be accessed under Windows, yet not to be visible. Such files or directories are assigned the DOS attribute <i>hidden</i>.</p> <p>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 ?.</p> <p>As an example, <i>./.*/test/</i> hides all files and directories the names of which begin with a dot, or which are called test.</p> <p>Note</p> <p>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.</p>
NT ACL support	<p>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.</p> <p>If this option is not set, existing POSIX ACLs are effective but not shown under Windows, and consequently cannot be changed under Windows.</p>
Users with write access	Only the users and groups listed here have write permission for the corresponding share.
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.
Inherit owner	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.

Attribute	Description
Inherit permissions	When activating this option, for each file or directory created in this share, the UNIX permissions of the superior directory will automatically be adopted.

Table 10.5. 'Samba custom settings' tab (advanced settings)

Attribute	Description
Custom share settings	<p>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 <code>man smb.conf</code>. 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.</p> <p>Caution</p> <p>The definition of extended Samba settings is only necessary in very special cases. The options should be thoroughly checked before setting since they might have security-relevant effects.</p>

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** or 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 authorisations might be changed according to the Samba settings on this tab.

Table 10.6. '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 authorisations 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 10.7. 'Samba performance' tab (advanced settings)

Attribute	Description
Locking	<p>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.</p> <p>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.</p>
Blocking locks	<p>Clients can send a lock request with a time limit for a certain area of an open file.</p> <p>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.</p>
Strict locking	<p>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.</p> <p>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.</p>
Oplocks	If this option is activated, Samba will use so-called <i>opportunistic locks</i> . 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.

Attribute	Description
Level 2 oplocks	<p>When activating this option, Samba will support an extended form of Oplocks, the so-called <i>opportunistic read-only locks</i> or <i>Level 2 Oplocks</i>. 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.</p> <p>It is recommended to activate this option to speed up access to files which are normally not written to (e.g. programs / executable files).</p> <p>Note</p> <p>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.</p>
Fake oplocks	<p>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.</p> <p>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.</p>
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 <i>manual</i> , <i>documents</i> , <i>programs</i> , and <i>disable</i> .

Table 10.8. '(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.

10.3. 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 10.2) 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 utilise 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 *zufb* 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 zufb
```

This reference will be displayed on every client capable of MSDFS (e.g. Windows 2000 and 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 spelt 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'.

Chapter 11. Print services

11.1. Introduction	165
11.2. Installing a print server	165
11.3. Setting the local configuration properties of a print server	166
11.4. Creating a printer share	166
11.5. Creating a printer group	169
11.6. Administration of print jobs and print queues	170
11.7. Generating PDF documents from print jobs	171
11.8. Mounting of print shares in Windows clients	171
11.9. Integrating additional PPD files	172

11.1. Introduction

[Feedback](#) 

Univention Corporate Server includes a print system, which can also be used to realise 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 the Univention Management Console, see Section 11.6.

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 colour, 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 11.9.

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 neighbouring 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.

Print shares from Windows systems can also be integrated in the CUPS print server, see Section 11.4.

11.2. Installing a print server

[Feedback](#) 

A print server can be set up on all domain controllers and member servers by selecting the installer component *Print server (CUPS)* or via subsequent installation of the ***univention-printserver*** package. After installation of the package, *univention-run-join-scripts* must be run.

11.3. 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.

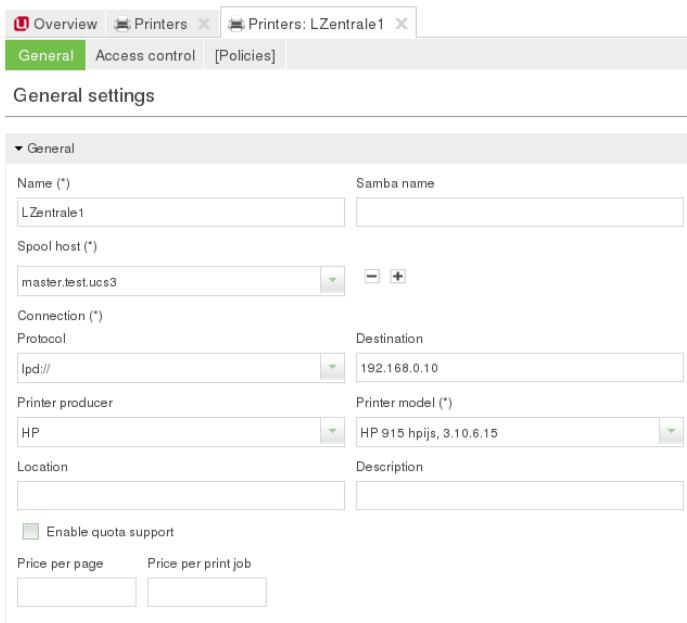
11.4. Creating a printer share

[Feedback](#)

Print shares are administrated in the *Printers* module of the Univention Management Console with the **Printer share** object type.

General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

Figure 11.1. Creating a printer share



The screenshot shows the 'General' tab of a printer share configuration page. The tabs at the top are 'Overview', 'Printers' (selected), and 'Printers: LZentrale1'. The 'General' tab is active. The form fields include:

- Name (*)**: LZentrale1
- Samba name**: (empty)
- Spool host (*)**: master.test.ucs3
- Connection (*)**: Protocol: lpd://; Destination: 192.168.0.10
- Printer producer**: HP
- Printer model (*)**: HP 915 hpijs, 3.10.6.15
- Location**: (empty)
- Description**: (empty)
- Enable quota support**: (checkbox)
- Price per page**: (empty)
- Price per print job**: (empty)

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 11.1. '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.
Spool host (*)	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

Attribute	Description
	<p>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.</p> <p>Only domain controllers and member servers on which the univention-printserver package is installed are displayed in the list.</p>
Protocol and Destination (*)	<p>These two input fields specify how the print server accesses the printer:</p> <p>The following list describes the syntax of the individual protocols for the configuration of printers connected locally to the server:</p> <ul style="list-style-type: none"> • <code>parallel://<devicefile></code> Example: <code>parallel://dev/lp0</code> • <code>socket://<server>:<port></code> Example: <code>socket://printer_03:9100</code> • <code>usb://<devicefile></code> Example: <code>usb://dev/usb/lp0</code> <p>The following list describes the syntax of the individual protocols for the configuration of network printers:</p> <ul style="list-style-type: none"> • <code>http://<server>[:<port>]/<path></code> Example: <code>http://192.168.0.10:631/printers/remote</code> • <code>ipp://<server>/printers/<queue></code> Example: <code>ipp://printer_01/printers/xerox</code> • <code>lpd://<server>/<queue></code> Example: <code>lpd://10.200.18.30/bwdraft</code> <p>The <code>cups-pdf</code> protocol is used for integrating a pseudo printer, which creates a PDF document from all the print jobs. The setup is documented in Section 11.7.</p> <p>The <code>file:/</code> 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.</p> <p>The <code>smb://</code> protocol can be used to mount a Windows print share. For example, to integrate the <code>laser01</code> printer share from Windows system <code>win01</code>, <code>win01/laser01</code> 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 <code>smb://win01/laser01</code>. No Windows drivers are used in this.</p>

Attribute	Description
	Independant 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 <i>Printer model</i> 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 11.9 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. 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. The print quota system needs to be installed for this.
Price per page	The user is charged the value given in this input field for every page printed. The incurred costs are summarised 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. 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 summarised 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. 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 11.2. '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.

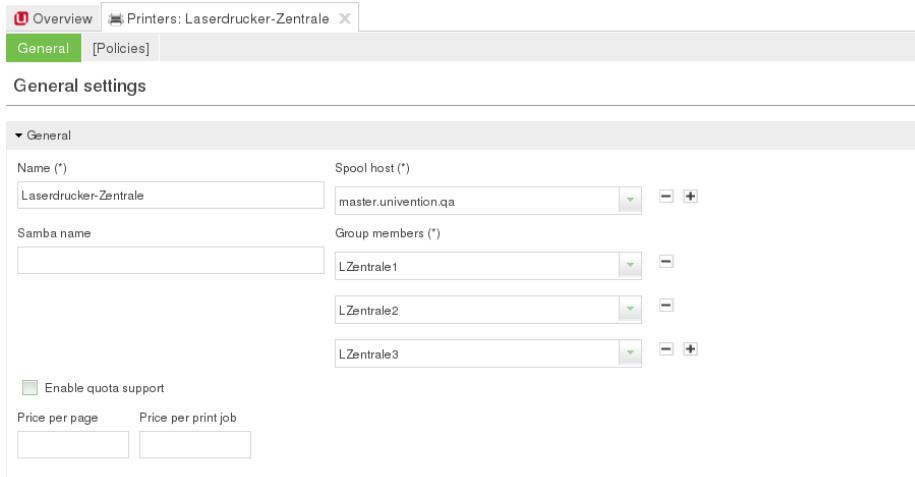
Attribute	Description
	This access control is useful for the management of printers spread across several locations, so that the users at location A do not see the printers of location B.
Allowed/denied users	This lists individual users for whom access should be controlled.
Allowed/denied groups	This lists individual groups for whom access should be controlled.

11.5. 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 utilisation is kept uniform.

Figure 11.2. 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 *Printers* module of the Univention Management Console with the **Printer share** object type.

General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

Table 11.3. '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.

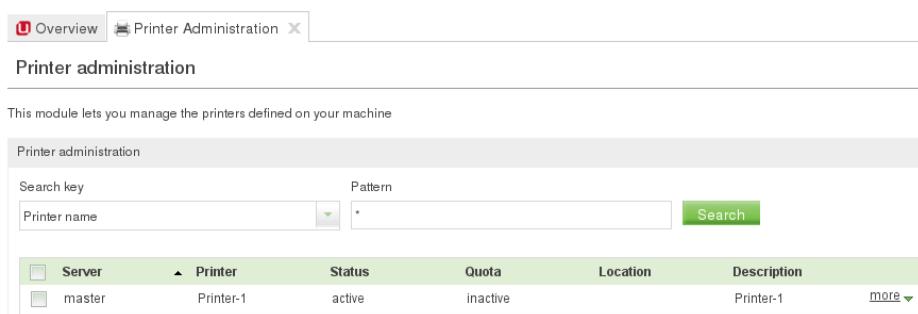
Attribute	Description
	The name may contain alphanumeric characters (i.e., uppercase and lowercase letters a to z and numbers 0 to 9) as well as hyphens and underscores. Other characters (including blank spaces) are not permitted.
Spool host (*)	A range of print servers (spoolers) can be specified here 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 be reached from Windows. Unlike the CUPS name (see <i>Name</i>), 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. 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. The print quota system needs to be installed for this.
Use Windows client driver	This option is documented in Section 11.8.
Price per page	The user is charged the value given in this input field for every page printed. The incurred costs are summarised 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 summarised 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.

11.6. Administration of print jobs and print queues

[Feedback](#)

The **Printer Administration** module of the Univention Management Console allows you to check the status of the connected printers, restart paused printers and remove print jobs from the queues on printer servers.

Figure 11.3. Printer administration



The screenshot shows the Univention Management Console with the 'Printer Administration' module selected. At the top, there is a navigation bar with tabs for 'Overview' and 'Printer Administration'. Below the navigation bar, the title 'Printer administration' is displayed. A descriptive text states: 'This module lets you manage the printers defined on your machine'. The main area features a search bar with fields for 'Search key' (containing 'Printer name') and 'Pattern' (containing '*'). Below the search bar is a table with the following columns: Server, Printer, Status, Quota, Location, and Description. One row is visible in the table, showing 'master' as the Server, 'Printer-1' as the Printer, 'active' as the Status, 'inactive' as the Quota, 'Printer-1' as the Location, and a link labeled 'more' under the Description column.

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.

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.

11.7. 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 11.4); 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**).

11.8. Mounting of print shares in Windows clients

[Feedback](#) 

The printer shares set up in the Univention Management Console can be added to Windows systems as network printers. The printer drivers need to be set up during the first access. To facilitate user access to the drivers, the drivers can also be stored on the server, as is described below. Alternatively, the option **Use Windows client driver** can also be activated on a printer share. This is necessary for Samba to accept print jobs from printer drivers only installed by users locally.

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 colour printer should be accessed, a driver for a Postscript-compatible colour 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.

The following steps make it possible to install a new printer driver on the printer server:

1. When logged on to Windows as **Administrator** you can open the printer server and the printer directory it contains.
2. Right clicking in a free area of the window opens the context menu where you can open the **Server properties** dialogue. Clicking on **Add** on the **Drivers** tab allows you to select a printer driver for the system architecture required and load this onto the server all via menus.

3. A right click on the printer will now run the **Properties** dialogue. The **Properties** tab contains a selection field for the drivers already installed. Clicking on **[OK]** assigns the driver to the printer and means that users privileges can access the printer without the need to install a driver.

11.9. 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 colour, 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 the 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 **LDAP directory** module of the Univention Management Console. 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 11.4. '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 11.4).
Driver	The path to the ppd file or to the <code>/usr/share/ppd/</code> directory. For example, if the <code>/usr/share/ppd/laserjet.ppd</code> should be used, <code>laserjet.ppd</code> must be entered here. gzip compressed files (file ending <code>.gz</code>) 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 12. Mail services

12.1. Introduction	173
12.2. Installation	174
12.3. Management of the mail server data	174
12.3.1. Management of mail domains	174
12.3.2. Assignment of e-mail addresses to users	175
12.3.3. Management of mailing lists	175
12.3.4. Management of mail groups	176
12.3.5. Management of shared IMAP folders	177
12.3.6. Mail quota	178
12.4. Spam detection and filtering	179
12.5. Identification of viruses and malware	180
12.6. Integration of Fetchmail for retrieving mail from external mailboxes	180
12.7. Configuration of the mail server	181
12.7.1. Configuration of a relay host for sending the e-mails	181
12.7.2. Configuration of the maximum mail size	181
12.7.3. Configuration of a blind carbon copy for mail archiving solutions	181
12.7.4. Configuration of soft bounces	181
12.7.5. Handling of mailboxes during e-mail changes and the deletion of user accounts	182
12.7.6. Distribution of an installation on several mail servers	182
12.8. Webmail and administration of e-mail filters with Horde	182
12.8.1. Login and overview	182
12.8.2. Web-based mail access	183
12.8.3. Address book	183
12.8.4. E-mail filters	184

12.1. Introduction

[Feedback](#) 

Univention Corporate Server provides mail services that users can access both via standard mail clients like Mozilla Thunderbird and via the webmail solution Horde.

Postfix is used for sending mails. In the basic installation, a Postfix configuration equipped for local mail delivery is set up on every UCS system. In this case, Postfix only accepts mails from the local server and they can also only be delivered to local system users. If e-mails also need to be sent to external addresses, a mail relay needs to be configured in this scenario (see Section 12.7.1).

The mail server component can be installed on server roles which realises complete mail transport via SMTP (see Section 12.2). The following functions are only available when the mail server component is used.

Postfix performs a validity test for incoming e-mails in the form of a search in the LDAP directory. That means that e-mails are only accepted for e-mail addresses defined in the LDAP directory or via an alias.

Following the optional spam and virus filtering, the received e-mails are transferred to the Cyrus server for the provision of e-mails via IMAP and POP3. The authentication is performed using the primary e-mail address, i.e., it must be entered as the user name in mail clients.

Cyrus 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/cyrus/pop` to `no`. The same applies to IMAP and the Univention Configuration Registry variable `mail/cyrus/imap`.

E-mails to foreign domains are sent directly to the responsible STMP server as standard. The location of the server is performed via the resolution of the MX record in the DNS. Mail sending can also be taken over by a relay host, e.g., on the Internet provider. This is described in the Section 12.7.1.

Spam mails are identified via the classification software SpamAssassin and sorted out by filter scripts (see Section 12.4). ClamAV is employed for the identification of viruses and other malware (see Section 12.5).

The UCS mail system does not offer any groupware functions such as shared calendars or invitations to appointments. There are further groupware systems based on UCS, which also offer integration into the UCS management system, for example: Kolab, Zarafa or Open-Xchange. More detailed information can be found on the Univention website [univention-technologypartners].

Version 4 of the Horde framework is provided for the web-based provision of e-mails (see Section 12.8)

The management of the user data of the mail server (e.g., e-mail addresses or mailing list) is performed in the Univention Management Console and is documented in the Chapter Section 12.3. The configuration of the mail server is performed via Univention Configuration Registry (see Section 12.7).

The UCS mail system offers the possibility of distributing users across several mail servers. The mail home servers described in the Section 12.7.6 are used for this purpose.

As soon as a user in the LDAP directory is assigned a primary e-mail address, a listener module creates an IMAP account on the user's mail home server.

Mail servers are configured using Univention Configuration Registry (see Section 12.7).

As the name of the mailbox is linked to the primary e-mail address and not the user name, a user can no longer access his old e-mails when changes are made to the primary e-mail address! If another user is assigned a previously used primary e-mail address, he receives access to the old IMAP structure of this mailbox.

12.2. Installation

[Feedback](#) 

The mail server can be set up in the Univention installer by selecting the *Mail server (Postfix, Cyrus IMAPd)* component. On an already installed system, the installation is performed using the **univention-mail-server** package.

The package 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 Cyrus server are stored in the `/var/spool/cyrus` directory. This directory should not be operated on a NFS share.

The webmail interface is provided in the Univention installer as *Webmailer (Horde4)* and can be subsequently installed with the **univention-horde4** package.

12.3. Management of the mail server data

[Feedback](#) 

12.3.1. Management of mail domains

[Feedback](#) 

A mail domain is a 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.

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.

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 *Mail* module of the Univention Management Console with the **mail domain** object type.

General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

The name of a mail domain may only be composed of lowercase letters, the figures 0-9, full stops and hyphens.

The mail domains registered on a mail server are automatically also saved in the Univention Configuration Registry variable `mail/hosteddomains`.

[Feedback](#) 

12.3.2. Assignment of e-mail addresses to users

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 12.3.1).

A user can be assigned two different types of e-mail addresses:

- The *primary e-mail address* is used for authentication on Postfix and Cyrus. 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. The domain part of the e-mail address must be registered in the Univention Management Console (see Section 12.3.1).
- 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 the Univention Management Console (see Section 12.3.1).

E-mail addresses are managed in the *Users* module of the Univention Management Console.

General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

The **primary e-mail address** is entered in the **General** tab in the **User account** submenu.

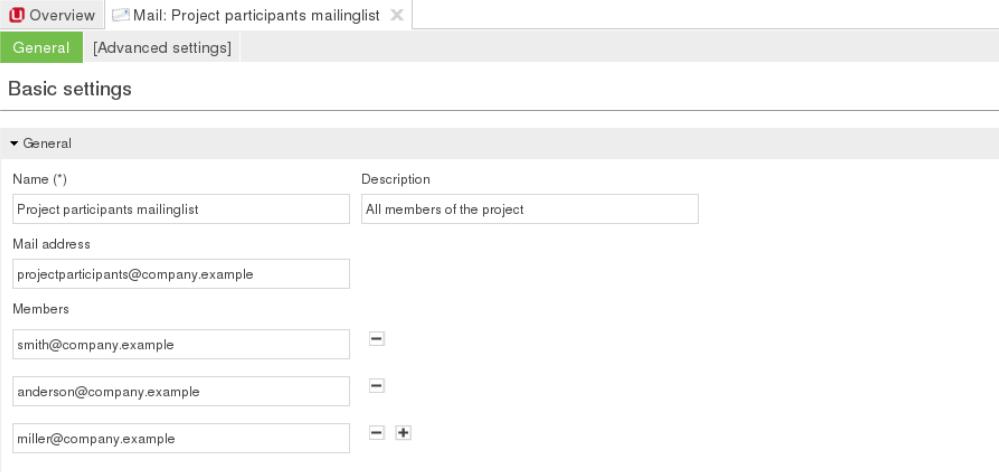
Alternative e-mail addresses can be entered under **Advanced settings -> Mail**.

[Feedback](#) 

12.3.3. Management of mailing lists

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 12.1. Creating a mailing list



Mail domains are managed in the *Mail* module of the Univention Management Console with the **Mailing list** object type.

General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

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 12.3.4), 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 authorised users/groups are allowed to send mails.

[Feedback](#)

12.3.4. Management of mail groups

There is the possibility of creating a mail group: This is where an e-mail address is assigned to a group. E-mails to this address are delivered to the primary address of each of the group members.

Mail groups are managed in the *Groups* module of the Univention Management Console.

General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

The 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 authorised users/groups are allowed to send mails.

12.3.5. 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 order is described as a shared IMAP folder.

Figure 12.2. Creating a shared IMAP folder

The screenshot shows the 'Mail' module interface. At the top, there are tabs for 'Overview' and 'Mail: umc.test'. Below these are two tabs: 'General' (which is selected) and 'Access Rights'. The main area is titled 'Basic settings' and contains a 'General' section. Within this section, there are fields for 'Name (*)' (set to 'Project IMAP folder'), 'Mail domain (*)' (set to 'umc.test'), 'Mail home server (*)' (set to 'master.umc.test'), 'Quota in MB' (set to '2048'), and 'E-Mail address' (set to 'project@umc.test').

Shared IMAP folders are managed in the *Mail* module of the Univention Management Console with the **IMAP Folder** object type.

General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

Table 12.1. 'General' tab

Attribute	Description
Name (*)	The name under which the IMAP folder is available in the e-mail clients.
Mail domain (*)	Every shared IMAP folder is assigned to a mail domain. The management of the domains is documented in the Section 12.3.1.
Mail home server (*)	An IMAP folder is assigned to a mail home server. Further information can be found in Section 12.7.6.
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.

Attribute	Description
	<p>The domain part of the e-mail address must be registered in the Univention Management Console (see Section 12.3.1).</p> <p>As soon as an e-mail address is entered for a folder, at least the IMAP rights <i>lrsp</i> are set for the user <i>anyone</i> so that the IMAP server can save e-mails in the IMAP folder.</p>

Table 12.2. 'Access rights' tab

Attribute	Description
Name (*)	<p>Access permissions based on users or groups can be entered here. Users are entered with their primary e-mail addresses; the groups saved in the Univention Management Console can be used as groups.</p> <p>The access permissions have the following consequences for individual users or members of the specified group:</p> <ul style="list-style-type: none"> No access No access is possible. The folder is not displayed in the folder list. Read The user may only perform read access to existing entries. Append Existing entries may not be edited; only new entries may be created. Write New entries may be created in this directory; existing entries may be edited or deleted. Post Sending an e-mail to this directory as a recipient is permitted. This function is not supported by all the clients. All Encompasses all permissions of <i>write</i> and also allows the changing of access permissions.

12.3.6. 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 by the *Mail quota* policy, which is managed under *Policies* in the *Users* module of the Univention Management Console.

General information on policy management can be found in Section 4.5.

The maximum size of the mailbox of a user is specified in the **Quota limit (MB)** field.

The user can be warned once a specified portion of the mailbox is attained and then receives a message with every incoming mail that his available storage space is almost full. This warning is shown by the e-mails clients and must thus be supported by them.

The administrator can enter the threshold in percent or remaining diskspace in kB:

- The threshold for when the warning message should be issued can be configured in the Univention Configuration Registry variable `mail/cyrus/imap/quotawarnpercent`. The value must be entered as a number between 0 and 100 without the percent sign.
- The Univention Configuration Registry variable `mail/cyrus/imap/quotawarnkb` is used to configure the threshold in kilobytes.

The quota is transferred to the quota settings of the Cyrus server during authentication on the mail server. The update interval is evaluated so that the quota settings are only updated once this time period has expired. This interval can be configured in minutes by Univention Configuration Registry variable `mail/cyrus/imap/quotainterval`.

No limit values are set for the IMAP storage space in the default setting. The use of mail quotas can be generally deactivated with the Univention Configuration Registry variable `mail/cyrus/imap/quota`.

12.4. Spam detection and filtering

[Feedback](#) 

Undesirable and unsolicited e-mails are designated as spam. The software SpamAssassin is 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.

Integration occurs via the `univention-spamassassin` package, which is 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 Cyrus, but rather in the *Spam* folder below it. 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 12.8.4).

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 its evaluation to the user's e-mail. The Bayes classification is controlled by the user himself, whereby e-mails not identified as spam can be placed in the *Spam* subfolder and a selection of legitimate e-mails copied into the *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.

12.5. 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 recognises 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.

12.6. 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).

Table 12.3. '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.

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 **System services** module of the

Univention Management Console. In that module the fetching can also be disabled (alternatively by setting the Univention Configuration Registry variable `fetchmail/autostart` to `false`).

12.7. Configuration of the mail server

[Feedback](#) 

12.7.1. 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-Relayhost> <username>:<password>
```

The command

```
postmap /etc/postfix/smtp_auth
```

must then be executed for this file to adopt the changes via Postfix.

12.7.2. 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 12.8) is used, the Univention Configuration Registry variables `php/limit/file-size` 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.

12.7.3. 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. As standard the variable is not set. If there is no mailbox for this address, one will be created automatically.

Postfix must then be restarted.

12.7.4. 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.

Handling of mailboxes during e-mail changes and the deletion of user accounts

12.7.5. 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/cyrus/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/cyrus/mailbox/delete` can be used to configure, whether the IMAP mailbox is also deleted when a user account is deleted. In the basic setting, the mailboxes are kept when a user account is deleted.

12.7.6. 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 12.3.5) are assigned to a mail home server.

The Cyrus Murder expansion is provided for the implementation of a mailbox cluster.

12.8. Webmail and administration of e-mail filters with Horde

[Feedback](#) 

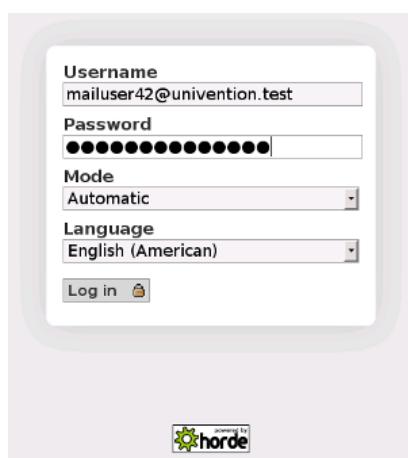
UCS integrates a number of applications from the Horde framework for web access to e-mails and web-based administration of server-side e-mail filter rules based on SIEVE.

12.8.1. Login and overview

[Feedback](#) 

The Horde login mask is linked on the system start page under **Horde4 web client** and can be opened directly at `http://SERVERNAME/horde4/login.php`.

Figure 12.3. Login on Horde 4



The screenshot shows the Horde 4 login interface. It consists of a form with the following fields:

- Username:** mailuser42@univention.test
- Password:** (Redacted)
- Mode:** Automatic
- Language:** English (American)

Below the form is a "Log in" button with a small icon. At the bottom of the screen is the Horde logo.

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.

The left half of the screen contains three menu points (**Mail**, **Filters** and **Address Book**), with which you can switch between the individual modules.

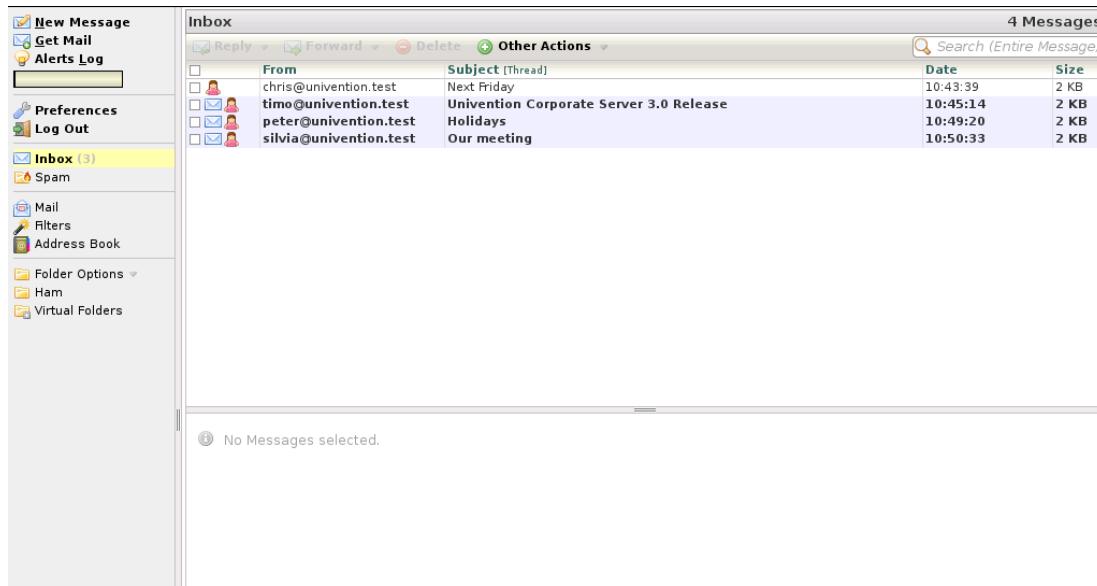
The user can personalise Horde under **Preferences**.

[Feedback](#)

12.8.2. Web-based mail access

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 12.4. Web mail (Inbox)

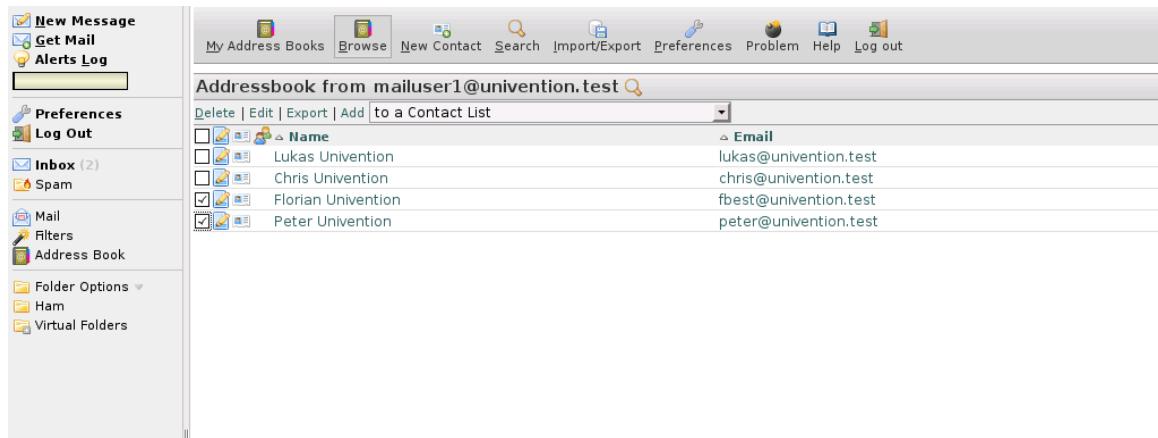


Horde differentiates between two types of deletion: an e-mail deleted with **Delete** is not definitively removed from the server, but only marked as to be deleted. It can be restored with **Undelete** and made available again as long as it is not removed from the IMAP server with **Purge Deleted**.

[Feedback](#)

12.8.3. Address book

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 12.5. Address book for webmail


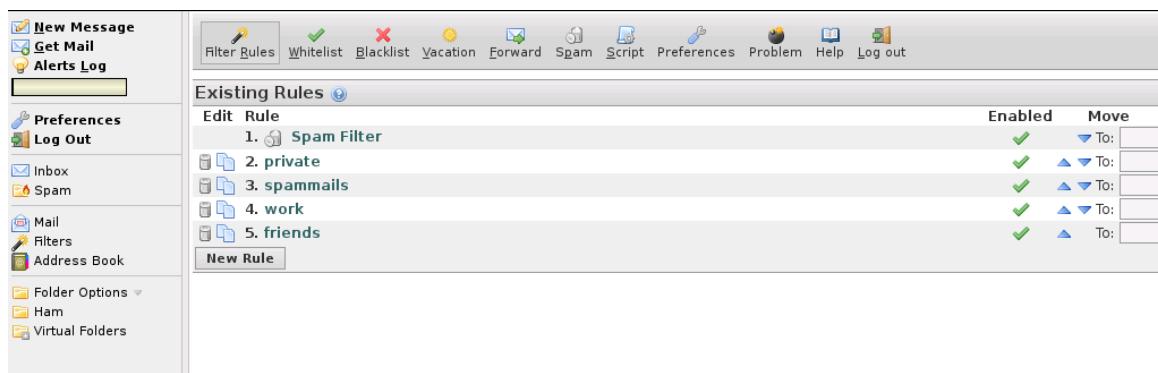
Contact information found using the simple or advanced search can then be copied into individual address books and edited there. New contacts can be entered via the **New Contact** menu item. Personal address books can also be created via **My Address Books**.

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.

12.8.4. E-mail filters

[Feedback](#)

Cyrus 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 12.6. Filter management in Horde


Filters can be edited and expanded under **Filter Rules**. 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 threshhold. 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 13. Infrastructure monitoring with Nagios

13.1. Introduction and structure	187
13.2. Installation	188
13.2.1. Preconfigured Nagios checks	189
13.3. Configuration of the Nagios monitoring	191
13.3.1. Configuration of a Nagios service	191
13.3.2. Configuration of a monitoring time period	193
13.3.3. Assignment of Nagios checks to computers	194
13.3.4. Einbindung von manuell erstellten Konfigurationsdateien	196
13.4. Querying the system status via the Nagios web interface	196

13.1. 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.

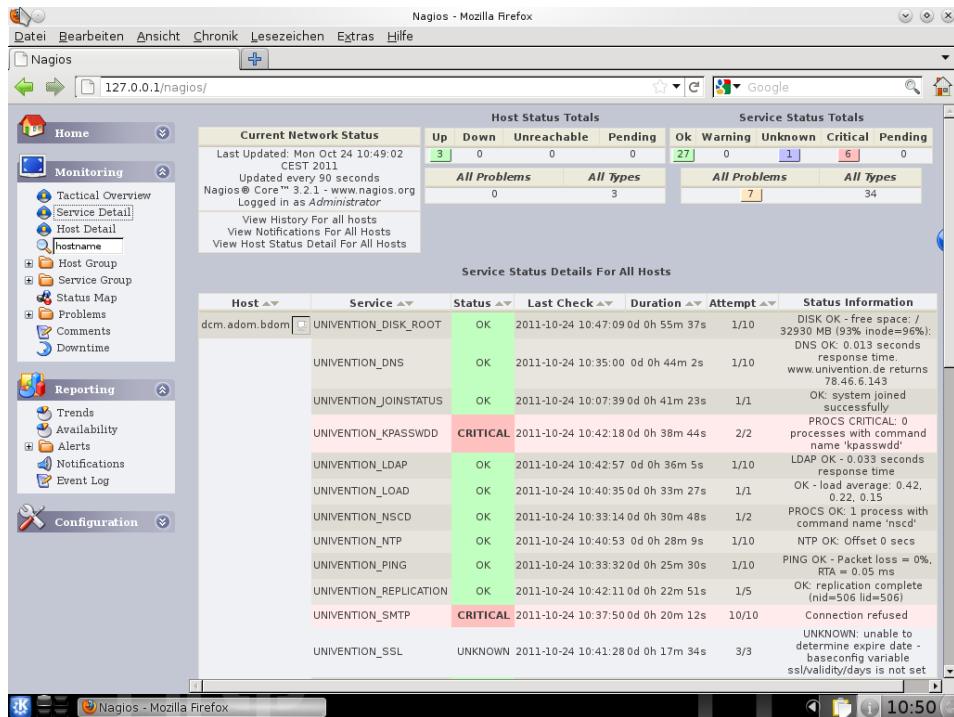
Nagios has a comprehensive collection of monitoring modules, the so-called Nagios plugins. In addition to polling system indicators (e.g., CPU and memory utilisation, free disk space), they also allow to test the availability and function of different services (e.g., SSH, SMTP, HTTP). Simple program 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 13.1. Nagios status webinterface


The screenshot shows the Nagios web interface in Mozilla Firefox. The main window displays the 'Host Status Totals' and 'Service Status Totals' summary. Below this, the 'Service Status Details For All Hosts' table provides detailed information for various services across different hosts. The table includes columns for Host, Service, Status, Last Check, Duration, Attempt, and Status Information.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
dcm.adom.bdom	UNIVENTION_DISK_ROOT	OK	2011-10-24 10:47:09 0d 0h 55m 37s	1/10		DISK OK - free space: / 32930 MB (95% inode=96%); DNS OK: 0.013 seconds
	UNIVENTION_DNS	OK	2011-10-24 10:35:00 0d 0h 44m 2s	1/10		www.univention.de returns 78.46.6.143
	UNIVENTION_JOINSTATUS	OK	2011-10-24 10:07:39 0d 0h 41m 23s	1/1		OK: system joined successfully
	UNIVENTION_KPASSWDD	CRITICAL	2011-10-24 10:42:18 0d 0h 38m 44s	2/2		PROCS CRITICAL: 0 processes with command name 'kpasswdd'
	UNIVENTION_LDAP	OK	2011-10-24 10:42:57 0d 0h 36m 5s	1/10		LDAP OK - 0.033 seconds response time
	UNIVENTION_LOAD	OK	2011-10-24 10:40:35 0d 0h 33m 27s	1/1		OK - load average: 0.42, 0.22, 0.15
	UNIVENTION_NSCD	OK	2011-10-24 10:33:14 0d 0h 30m 48s	1/2		PROCS OK: 1 process with command name 'nscd'
	UNIVENTION_NTP	OK	2011-10-24 10:40:53 0d 0h 28m 9s	1/10		NTP OK: Offset 0 secs
	UNIVENTION_PING	OK	2011-10-24 10:33:22 0d 0h 25m 30s	1/10		PING OK - Packet loss = 0%, RTA = 0.05 ms
	UNIVENTION_REPLICATION	OK	2011-10-24 10:42:11 0d 0h 22m 51s	1/5		OK: replication complete (nid=506 lid=506)
	UNIVENTION_SMTP	CRITICAL	2011-10-24 10:37:50 0d 0h 20m 12s	10/10		Connection refused
	UNIVENTION_SSL	UNKNOWN	2011-10-24 10:41:28 0d 0h 17m 34s	3/3		UNKNOWN: unable to determine expire date - baseconfig variable ssl/validity/days is not set

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 NRPE 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 NRPE is provided by the *Nagios client* component, which is preinstalled on all UCS system roles. *Nagios client*

The Nagios configuration is performed in the Univention Management Console, the Nagios configuration files are automatically generated from the information stored in the LDAP directory.

13.2. Installation

[Feedback](#)

The Nagios server can be installed on any system role; the use of a domain controller system is recommended. The *univention-nagios-server* package must be installed for the commissioning. After the installation of the package, *univention-run-join-scripts* must be run.

The Nagios client can be installed on any system role. The *univention-nagios-client* package must be installed for the commissioning. After the installation of the package, the *univention-run-join-scripts* must be run. The Nagios client has been automatically installed with new installations since UCS 3.0.

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.

[Feedback](#) 

13.2.1. Preconfigured Nagios checks

During the installation, basic Nagios tests are set up automatically for UCS systems. The mounting of additional services is documented in the Section 13.3.1.

Nagios service	Description
UNIVENTION_PING	tests the availability of the monitored UCS system with the command <code>ping</code> . In the default setting, an error status is attained if the response time exceeds 50 ms or 100 ms or package losses of 20% or 40% occur.
UNIVENTION_DISK_ROOT	monitors how full the <code>/</code> 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 <code>www.univention.de</code> . If no DNS forwarder is defined for the UCS domain, this request fails. In this case, <code>www.univention.de</code> 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.
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 utilisation 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 recognises the creation of a <code>failed.1dif</code> 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.

Preconfigured Nagios checks

Nagios service	Description
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 <code>virsh</code> 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 <code>virsh</code> 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 customised 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 13.3.1.

The following Nagios services are only available on the respective Nagios client once additional packages have been installed (see Section 13.2):

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 <code>/dev/sda</code> . Corresponding Nagios services exist for the hard drives <code>sdb</code> , <code>sdc</code> and <code>sdd</code> .
UNIVENTION_RAID	tests the status of the software RAID via <code>/proc/mdadm</code> and returns CRITICAL if 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.

13.3. Configuration of the Nagios monitoring

The following settings can be performed in the Univention Management Console:

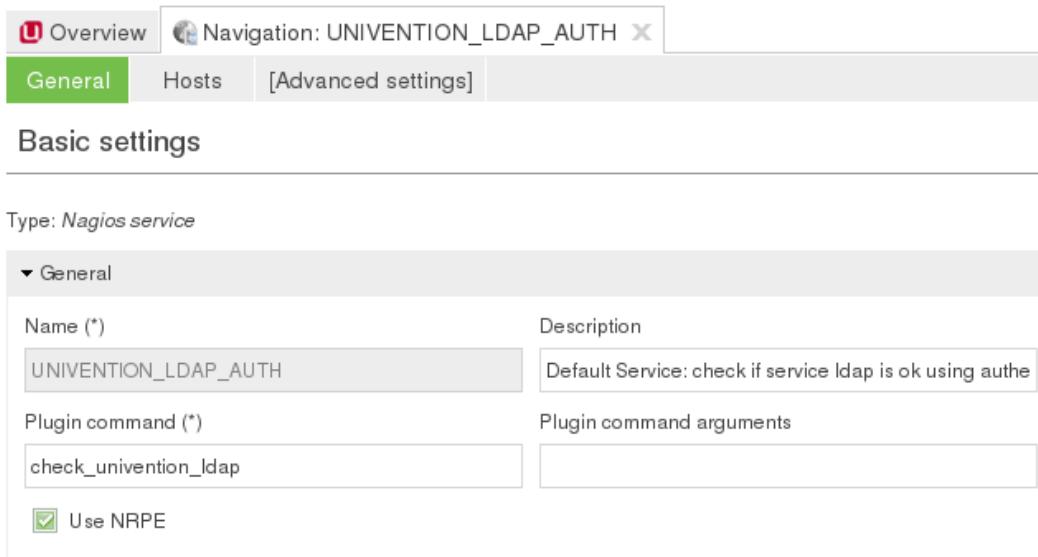
- All Nagios tests that can be assigned to a computer must be registered. This is performed via *Nagios service* objects, see Section 13.3.1.
- The assignment on which tests should be performed on a computer and which contact persons should be informed in the case of errors is performed on the respective 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 13.3.2.

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.

13.3.1. Configuration of a Nagios service

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.

Figure 13.2. Configuring a Nagios service



General	
Name (*)	Description
UNIVENTION_LDAP_AUTH	Default Service: check if service ldap is ok using auth
Plugin command (*)	Plugin command arguments
check_univention_ldap	
<input checked="" type="checkbox"/> Use NRPE	

Nagios services are administrated in the *Nagios* module of the Univention Management Console with the object type **Nagios service**.

General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

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.

Table 13.1. 'General' tab

Attribute	Description
Name	An unambiguous name for the Nagios service.

Attribute	Description
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 <code>/etc/nagios-plugins/config/</code> directory, e.g., <code>check_disk</code> .
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., <code>20%!10%!/home</code> .
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 <code>univention-nagios-client</code> package must be installed.

Table 13.2. '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 <i>OK</i> , 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 <i>OK</i> 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 <i>OK</i> status, the number of tests specified here is waited before the contact persons are notified. If the service reattains the <i>OK</i> status again before reaching the limit specified here, the internal counter is reset and there is no notification.
Check period	Note The time delay for a notification is arranged both according to the <i>maximum number of check attempts</i> and to the <i>retry check interval</i> . At a <i>retry check interval</i> of two minutes and a <i>maximum number of check attempts</i> of 10, the first notification is performed after 20 minutes.

Table 13.3. '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.

Attribute	Description
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.
	Note
	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 <i>WARNING</i> (see Section 13.1).
Notify if service state changes to CRITICAL	Configures whether a notification is sent when the service status changes to <i>CRITICAL</i> (see Section 13.1).
Notify if service state changes to UNREACHABLE	If a computer object is subordinate to another object (see Section 13.3.3), 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 13.4. 'Hosts' tab

Attribute	Description
Assigned hosts	The service test is performed for/on the computers assigned here.

[Feedback](#)

13.3.2. Configuration of a monitoring time period

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 *Nagios* module of the Univention Management Console with the **Nagios time period** object type:

General information on the operation of the domain management modules of the Univention Management Console and on the adding, editing, searching for and deleting of objects can be found in Section 4.2.

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.

Nagios time period	Description
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.

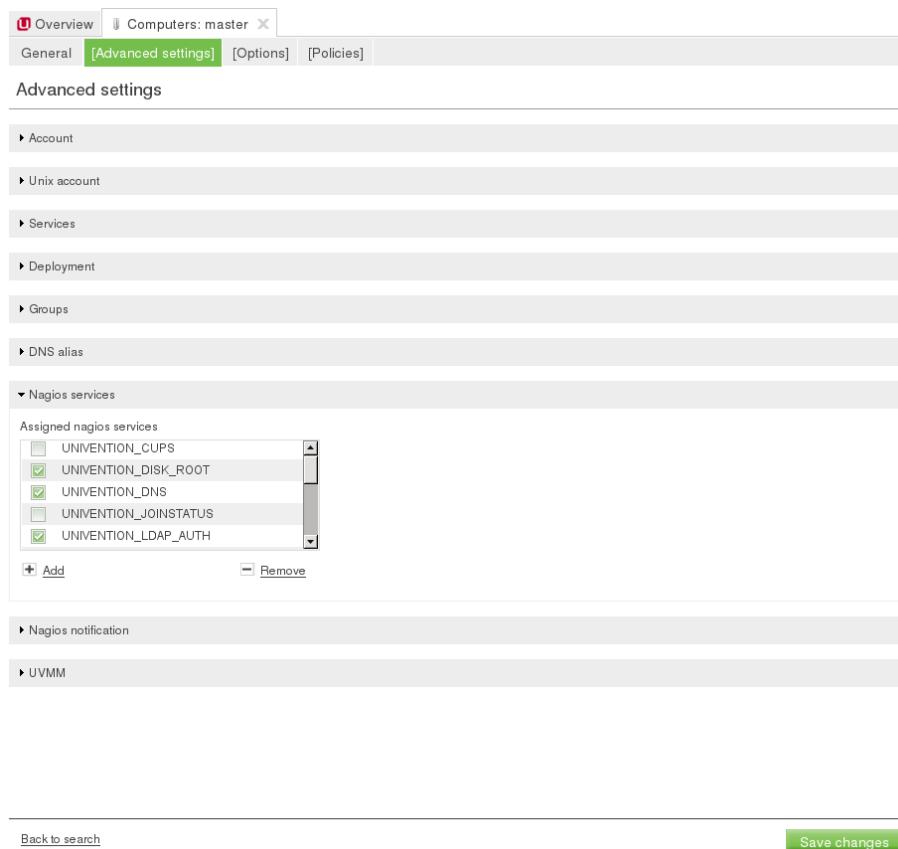
Table 13.5. 'General' tab

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.

13.3.3. 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 tabs available, which can be used to assign the Nagios services conveniently among other things.

Figure 13.3. Assigning Nagios checks to a host


The screenshot shows the 'Advanced settings' tab for a computer named 'master'. The 'Assigned nagios services' section contains five entries: UNIVENTION_CUPS, UNIVENTION_DISK_ROOT, UNIVENTION_DNS, UNIVENTION_JOINSTATUS, and UNIVENTION_LDAP_AUTH. The first four are checked, while the last one is unchecked. There are 'Add' and 'Remove' buttons below the list.

Table 13.6. 'Nagios services' tab (advanced settings)

Attribute	Description
Assigned Nagios services	All the Nagios services that are checked for the current computer are listed here. Parallel to this, the assignment of computers on the Nagios service object is also possible.

Table 13.7. '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 <i>root</i> 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
	<p>Note</p> <p>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.</p>

13.3.4. Einbindung von manuell erstellten Konfigurationsdateien

[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/nagios3/conf.local.d/` directory. The added configuration files are only taken into account after the next restart of the server.

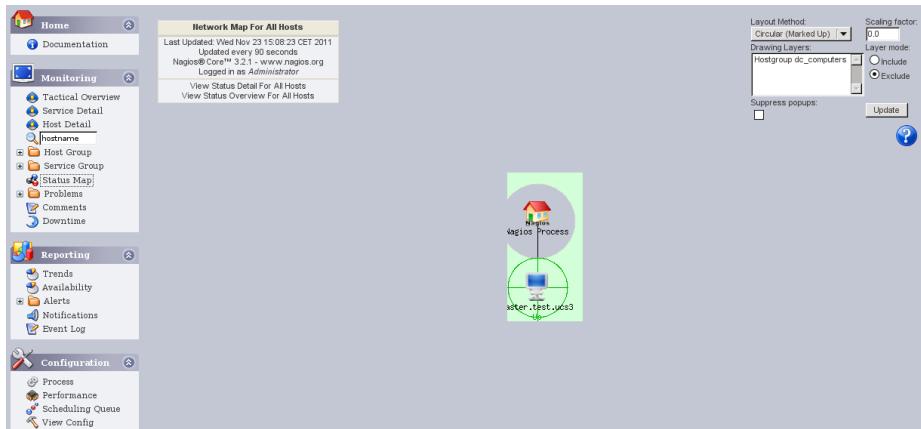
Expansions to the NRPE configurations can be stored in the `/etc/nagios/nrpe.local.d/` directory. Changes are only applied after the next restart of the Nagios NRPE Daemon.

13.4. Querying the system status via the Nagios web interface

[Feedback](#)

The Nagios interface can be accessed at `https://<ip-oder-fqdn>/nagios/`.

Figure 13.4. 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 authorised to log in.

Chapter 14. Virtualization

14.1. Introduction	197
14.2. Installation	198
14.3. Image files of virtual machines	198
14.4. Accessing the default storage pool through a file share	199
14.5. CD/DVD/floppy drives in virtual machines	200
14.6. Network interfaces in virtual instances	200
14.7. Paravirtualization/virtIO drivers for Microsoft Windows systems	201
14.7.1. Installation of the GPLPV drivers for Xen instances	201
14.7.2. Installation of the virtIO drivers for KVM instances	202
14.8. Snapshots	202
14.9. Migration of virtual instances	203
14.9.1. Migration of virtual machines from failed virtualization servers	203
14.10. Managing virtual machines with the Univention Management Console	203
14.10.1. Operations (Starting/stopping/suspending/deleting/migrating/cloning virtual machines)	204
14.10.2. Creating a virtual instance	205
14.10.3. Modifying virtual machines	206

14.1. Introduction

[Feedback](#) 

UCS Virtual Machine Manager (UVMM) is a management system for virtualization servers and virtual machines. It offers the possibility of monitoring all the virtualization servers registered in the UCS domain and administrating the virtual instances on these systems.

A Univention Management Console module forms the management interface. All virtualization servers are then administrated from this management system, see Section 14.10).

Section 14.2 describes the installation of the management system and the virtualization servers and the functions of the Univention Management Console module.

In principle, the virtualized systems can run arbitrary operating systems. This mode of operation is called *fully virtualized* systems.

Both virtualization technologies Xen and KVM are equally supported by UCS Virtual Machine Manager. However, the technologies have different advantages and disadvantages depending on the host systems and hardware used. For example, KVM requires virtualization support in the central processing unit, while Xen can also virtualize (within limits) systems without support from the hardware. More detailed information can be found on their respective websites: <http://www.linux-kvm.org/> and <http://www.xen.org/>.

KVM and Xen each offer interfaces to provide the virtualized systems with direct access to the resources of the virtualization server. This considerably improves performance.

In Xen, this technology is referred to as *paravirtualization*. KVM offers the *virtIO* interface. This allows network and storage devices a direct connection to the KVM resources. This technology is comparable to paravirtualization. When menu points in UVMM refer to paravirtualization, this also includes virtIO.

UVMM supports both full virtualization and paravirtualization for virtualized systems. Using paravirtualization/virtIO is recommended.

Current Linux systems support paravirtualization as standard. virtIO and Xen paravirtualization drivers for Microsoft Windows are included in UCS. The installation is documented in Section 14.7.

Univention Wiki (<http://wiki.univention.de/>) includes a step-by-step quickstart guide [uvmm-quickstart] and further technical documentation as well as howtos [uvmm-technical-details]. (Currently only available in German)

14.2. Installation

[Feedback](#) 

UCS Virtual Machine Manager comprises three different packages. They can all be selected directly when installing the UCS system or alternatively installed subsequently via Univention Management Console.

Virtual Machine Manager (UVMM) (Package: univention-virtual-machine-manager-daemon)

This package must be installed on the management system. In doing so, both an additional service and the Univention Management Console module are set up. This package should be installed on a domain controller in the UCS domain. The installation on member servers requires additional configuration steps, which are documented in [uvmm-technical-details]. If UVMM is installed on a slave domain controller or backup domain controller, the package *univention-virtual-machine-manager-schema* needs to be installed on the master domain controller beforehand.

Xen virtualization server (Package: univention-virtual-machine-manager-node-xen)

This package must be installed on each system which is to be used as a Xen-based virtualization server.

KVM virtualization server (Package: univention-virtual-machine-manager-node-kvm)

If KVM is to be used for the virtualization, this package should be installed on the virtualization servers.

The two packages for the virtualization servers register the service in the LDAP directory. Additionally, a particular kernel is installed on Xen systems, which is necessary to allow Xen to be used.

When installing the virtualization servers, only one virtualization technology should be used per server.

Additionally, the architecture must also be taken into account during installation of the virtualization servers. 64 bit systems can only be virtualized on UCS systems where are installed using the amd64 architecture. A 64-bit system (amd64) is recommended for use as the virtualization server.

14.3. 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. This technique is a prerequisite for efficiently managing snapshots of virtual machines.

Alternatively, you can also access a hard drive image in *Simple format (raw)*. Snapshots can only be created when using hard drive images in *Extended format*.

Only the *Simple format* is available on Xen systems.

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 synchronised 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, synchronisation 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 synchronisation 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 resaved with UCS 3.1 the standard value is implicitly *write-through* again, but new VMs are explicitly saved with *none*.

Xen uses its own system called Tapdisk, which bypasses the guest's page cache similarly to *none*.

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.

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 [uvmm-technical-details].

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 Chapter 13.

[Feedback](#) 

14.4. Accessing the default storage pool through a file share

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.

¹Instead, 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: Der Rechnername des UVMM-Servers
 - 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.

14.5. 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 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.

14.6. Network interfaces in virtual instances

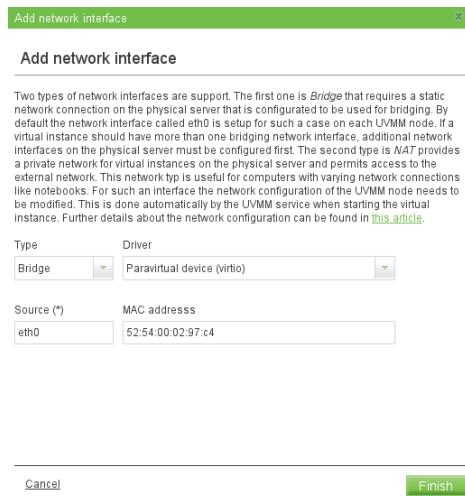
[Feedback](#) 

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 192.168.122.0/24 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 14.1. Adding a virtual network interface



The UVMM servers are already preconfigured for bridging and NAT. However, there are restrictions for bridged network cards. On the UVMM servers, the physical network card to which the standard route is set, is converted to a bridge in the default setting. If additional network cards are integrated in the server, these are not adapted accordingly. If several bridge network cards are required in a virtual instance, an additional network card must be configured as a bridge on the server in advance. If a bridge is used, the **Source** of the network interface used can be selected.

NAT network cards are only restricted by the IP addresses available in the 192.168.122.0/24 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.

[Feedback](#)

14.7. Paravirtualization/virtIO drivers for Microsoft Windows systems

virtIO and Xen paravirtualization drivers for Microsoft Windows are included in UCS. Installing the respective KVM or Xen packages provides appropriate images, which can be added in the drive settings of a virtual machine. The images are added to the storage pool specified with the Univention Configuration Registry variable `uvmm/pool/default/path`:

- On Xen virtualization servers an ISO image named *Xen Windows drivers (gplv 308)* is provided, which contains the GPLV virtualization driver for Microsoft Windows.
- On KVM virtualization servers an ISO and a floppy image named *KVM Windows drivers (virtio 1.1.16)* are provided, which contains the virtIO virtualization driver for Microsoft Windows.

[Feedback](#)

14.7.1. Installation of the GPLV drivers for Xen instances

[Feedback](#)

The GPLV driver is a open source driver for Microsoft Windows, which enables Windows DomU systems virtualized in Xen to efficiently access the network and storage resources of the Xen Dom0. This provides a significant performance and reliability gain over the emulated standard devices.

Univention provides the GPLV drivers signed with a Software Publishers Certificate obtained from the GlobalSign CA.

There are different MSI installer packages for the various Windows releases, which can be started with a simple double click. The **typical** installation variant should cover most use cases. The Windows installations need to be updated to the current service packs before installing the GPLPV drivers. E.g., using the GPLPV driver on Windows XP w/o SP3 is not possible.

Table 14.1. Windows virtualization drivers

File name of driver package	Windows version
gplpv_2000_signed_0.11.0.308.msi	Windows 2000
gplpv_2003x32_0.11.0.308.msi	Windows 2003 (32 Bit)
gplpv_2003x64_0.11.0.308.msi	Windows 2003 (64 Bit)
gplpv_Vista2008x32_0.11.0.308.msi	Windows Vista (32 Bit)
gplpv_Vista2008x32_0.11.0.308.msi	Windows 2008 (32 Bit)
gplpv_Vista2008x64_0.11.0.308.msi	Windows Vista (64 Bit)
gplpv_Vista2008x64_0.11.0.308.msi	Windows 2008 (64 Bit)
gplpv_XP_0.11.0.308.msi	Windows XP (32 Bit)

After successful installation and a reboot *Xen Net Device Driver* and *Xen Block Device Driver* can be found in the device manager.

14.7.2. 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 floppy drive needs to be setup in the drive settings with the image *virtio 1.1.16.vfd* assigned.
- The hard disk drive has to be edited in the **Devices** menu in UVMM and the checkbox **Paravirtual drive** must be ticked.
- 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.

14.8. 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 14.10.2) directly in the hard drive image file.

14.9. Migration of virtual instances

UVMM offers 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 14.2. 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 [uvmm-technical-details].

14.9.1. Migration of virtual machines from failed virtualization servers

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 unaccessible 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.

14.10. Managing virtual machines with the Univention Management Console

The Univention Management Console 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 (Xen or KVM), however they may vary slightly depending on the

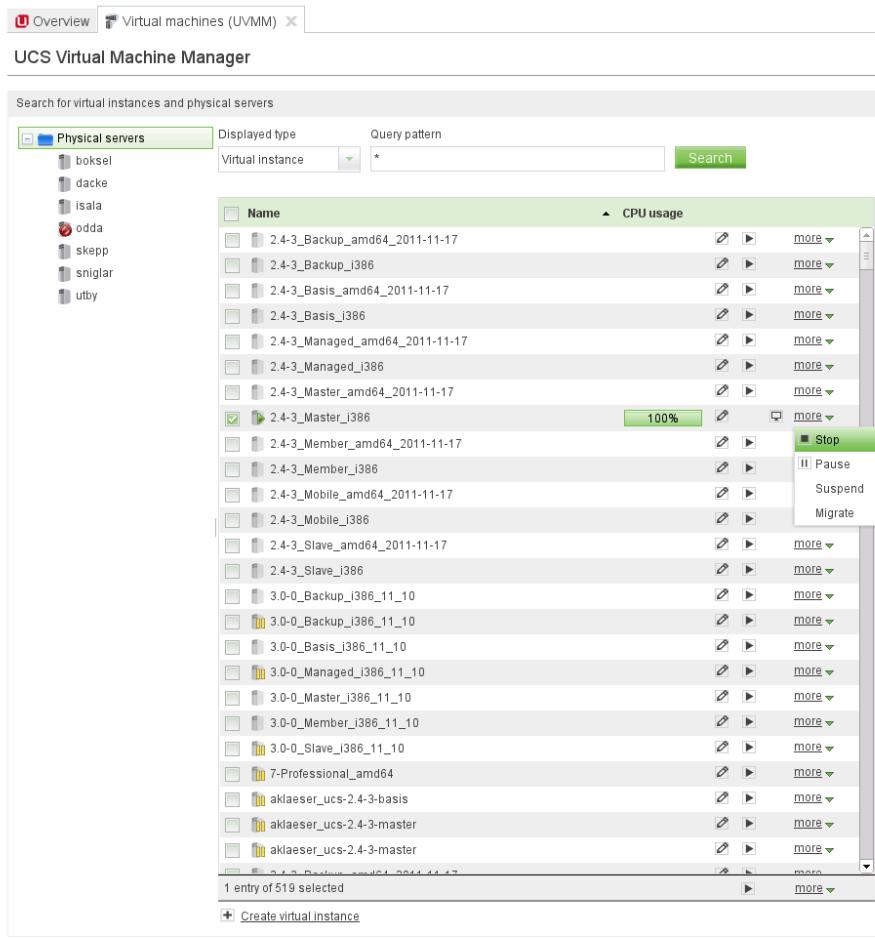
Operations (Starting/stopping/suspending/deleting/migrating/cloning virtual machines)

hypervisor in use. The items that must be observed are illustrated in the following section on the description of the functions.

14.10.1. Operations (Starting/stopping/suspending/deleting/migrating/cloning virtual machines)

[Feedback](#) 

Figure 14.3. Overview of virtual machines



In the main dialog 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 this 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).

Instances created via UVMM are turned off in the initial status. 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 stylised screen opens a connection with a Java-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 14.9.

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 14.9.

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!

[Feedback](#) 

14.10.2. Creating a virtual instance

Virtual machines can be created with the assistant in a few steps in UVMM by clicking on **Create virtual instance**.

The input mask **Physical server** can be used to select on which virtualization server the virtual machine should be created. The selection of the **Profile** specifies some of the basic settings for the virtual instance (e.g., a name prefix, no. of CPUs, RAM and whether the direct access per VNC should be activated).

The existing UVMM profiles are stored in the LDAP directory and can also be edited there. The profiles can be found in the **LDAP directory** section of the Univention Management Console in the container *cn=Profiles,cn=Virtual Machine Manager*. Additional profiles can also be added there.

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 14.3.

Clicking **Finish** concludes the creation of the virtual machine.

14.10.3. Modifying virtual machines

[Feedback](#)

In the overview list, a virtual machine can be edited by clicking on the icon with the stylized pen.

Figure 14.4. Modifying the settings of a DVD drive

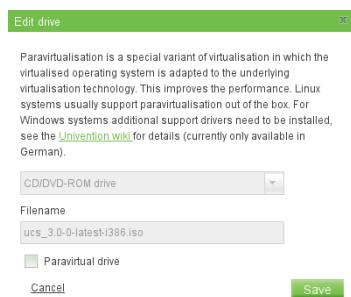


Table 14.2. 'General' tab

Attribut	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. <i>mail server</i> or its 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 14.3. An introduction to the supported network card settings can be found in the Section 14.6.

Drives lists all existing drives, the image files used, their size and the assigned storage pools. One can click on the stylised 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 utilisation 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 14.8.

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 stylised 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 2.4-2*. In addition to the description the time is saved when the snapshot is created.

The settings of a virtual machine can only be changed if it is turned off.

Table 14.3. '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 and Xen 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.
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.
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 can 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.
Direct access (VNC)	Defines whether VNC access to the virtual machine is available. If the option is activated, the UMC module can be used to start a VNC program directly. The VNC URL is displayed in a tool tip. A Java VNC program is used for this in the default setting.
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.

Chapter 15. Data backup with Bacula

15.1. Introduction	209
15.2. Installation	210
15.3. Configuration of the backup components	210
15.3.1. Directory Daemon	210
15.3.2. Storage	211
15.3.3. File Daemon	211
15.3.4. Bacula Console	212
15.3.5. Firewall adjustments	212
15.4. Configuration of the backup (interval, data, etc.)	212
15.5. Administration via the Bacula console	213
15.6. Backup of the catalog database	213
15.7. Further information	214

15.1. Introduction

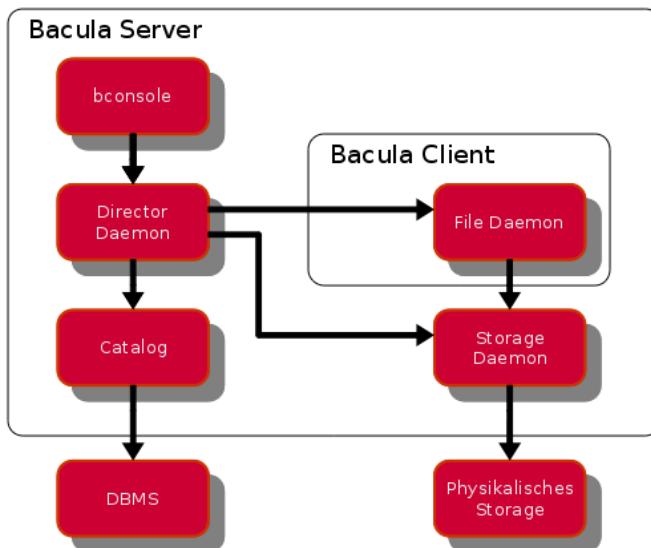
[Feedback](#) 

Bacula is a network-enabled data backup solution with a client/server architecture. It allows data backup and restore in heterogeneous environments.

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 15.1. Bacula Schema


15.2. Installation

[Feedback](#)

In this documentation it is assumed that the *director daemon*, *storage daemon* and *catalog* are present on one 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.

15.3. 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.

15.3.1. 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 the local host, 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 = 192.168.100.125
}
```

15.3.2. 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 = 192.168.100.125
}
```

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"
}
```

15.3.3. 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 15.3.1). A client password must be set for every system. In addition, the *FDAAddress* 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
    FDAAddress = 192.168.100.125
}
```

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 = 192.168.100.125
```

```
FDPort = 9102
Catalog = MyCatalog
Password = "client-password"
File Retention = 30 days
Job Retention = 6 months
AutoPrune = yes
}
```

15.3.4. 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 15.3.1):

```
Name = localhost-dir
DIRport = 9101
address = 192.168.100.125
Password = "master-dir-password"
```

15.3.5. 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.

15.4. 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 utilisation 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.

15.5. 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.

15.6. 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"
```

Further information

```
    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.

15.7. 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/3.0.x-manuals/en/install/install.pdf>
- <http://en.wikipedia.org/wiki/Bacula>

Bibliography

- [ucs-handbuch] Univention GmbH. 2012. *Univention Corporate Server - Manual for users and administrators*. <http://docs.univention.de/manual-3.1.pdf>.
- [ucs-dokumentationen] Univention GmbH. 2012. *Documentation on Univention products*. <http://www.univention.de/en/download-and-support/documentation/documentation-on-univention-products/>.
- [wiki-samba4] Univention GmbH. 2012. *Univention Wiki - Samba in UCS 3.0*. http://wiki.univention.de/index.php?title=Samba_in_UCS_3.0%2Fen.
- [locales] Debian Project. 2012. *Locale - Debian Wiki*. <http://wiki.debian.org/Locale>.
- [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>.
- [wiki-samba-update] Univention GmbH. 2012. *Univention Wiki - Update to UCS 3.0 Samba 4*. http://wiki.univention.de/index.php?title=Update_to_UCS_3.0_Samba_4.
- [wiki-ad-takeover] Univention GmbH. 2012. *Univention Wiki - UCS 3.1 Univention AD Takeover*. http://wiki.univention.de/UCS_3.1_Univention_AD_Takeover.
- [sdb-acls] Univention GmbH. 2011. *Univention Support Database - Wie können ACLs in UCS konfiguriert werden (currently only available in German)*. <http://sdb.univention.de/1042>.
- [sdb-sslchange] Univention GmbH. 2013. *Univention Support Database - Renewing the SSL certificate*. <http://sdb.univention.de/1183>.
- [univention-technologypartners] Univention GmbH. 2012. *Univention partners*. <http://www.univention.de/en/partner/technology-partners/>.
- [ext-doc-inst] Univention GmbH. 2012. *Extended installation documentation*. <http://docs.univention.de/installation-3.1.pdf>.
- [ext-doc-net] Univention GmbH. 2013. *Extended network management documentation*. <http://docs.univention.de/networks-3.1.pdf>.
- [uvmm-technical-details] Univention GmbH. 2012. *Univention Wiki - UVMM - Technische Details (currently only available in German)*. http://wiki.univention.de/index.php?title=UVMM_Technische_Details.
- [uvmm-quickstart] Univention GmbH. 2012. *Univention Wiki - UVMM - Quickstart Guide*. http://wiki.univention.de/index.php?title=UVMM_Quickstart-3.0%2Fen.
- [hardwarelist] Univention GmbH. 2012. *Univention Corporate Server - Compatible hardware*. <http://www.univention.de/en/products/ucs/product-information/compatible-hardware/>.
- [ec2-quickstart] Univention GmbH. 2011. *Univention Wiki - Amazon EC2 Quickstart*. http://wiki.univention.de/index.php?title=Amazon_EC2_Quickstart.
- [xenserver-installation] Univention GmbH. 2013. *Univention Wiki - Citrix XenServer*. http://wiki.univention.de/index.php?title=Citrix_Xen_Server.
- [ubuntu-integration] Univention GmbH. 2012. *Ubuntu Integration*. <http://wiki.univention.de/index.php?title=Ubuntu>.

[bacula-config-example] Univention GmbH. 2012. *Bacula configuration example*. http://wiki.univention.de/index.php?title=Bacula_configuration_example.

[ext-doc-computers] Univention GmbH. 2012. *Extended computer management documentation*. <http://docs.univention.de/computers-3.1.pdf>.