



Univention Corporate Server - Extended installation documentation

Release 5.0

Sep 13, 2023

The source of this document is licensed under GNU Affero General Public License v3.0 only.

CONTENTS

1 Using a UCS appliance	1
2 Creating a UCS appliance/cloud image	3
3 Profile-based installation	11
4 Bibliography	21
Bibliography	23
Index	25

USING A UCS APPLIANCE

In addition to the traditional installation, there is also the possibility of providing UCS via an appliance image. These appliance images can be used both for simple commissioning in a virtualization solution such as VMware and for providing a cloud instance.

Appliances can be created with minimal effort. This is described in *Creating a UCS appliance/cloud image* (page 3).

Whilst some of the settings can be preconfigured globally in the image, it is still necessary for the end user to make final adjustments to the configuration, e.g., to set the computer name or the domain used. For this reason, a basic system is installed for the appliance image and a component set up, which then allows the end user to finalize the configuration. Alternatively, the configuration can also be performed automatically without user interaction. This is described in *Automatic configuration of an appliance* (page 5).

The interactive configuration can be performed in two ways:

- A graphic interface starts on the system, in which the web browser Firefox is started in full-screen mode and automatically accesses the configuration URL. This option is particularly suitable for images in virtualization solutions.
- The configuration can also be performed directly via an external web browser. In this case, the system's IP address must be known to the user (e.g., if it has been notified to him in advance in the scope of the provision of a cloud image).

In the scope of the initial configuration, the user can change the following settings in the default setting:

- Selection of the language, time zone and keyboard layout
- Configuration of the network settings
- Setup of a new UCS domain or joining a UCS or Microsoft Active Directory domain
- Software selection of UCS key components. The user can install software from other vendors at a later point in time via the Univention App Center.

CREATING A UCS APPLIANCE/CLOUD IMAGE

This section describes how to set up an appliance based on UCS 5.0. This type of appliance can also be used to provide preconfigured instances as a cloud service provider. The creation of images for typical virtualization solutions is another possible application scenario, see *Providing an image for VMware/VirtualBox* (page 5).

2.1 Performing the basic installation

The basic installation is performed using the standard UCS installer. Further information on the individual options can be found in the UCS manual. The installation should be performed in a virtualization solution. In this example, the installation is performed in QEMU. A `qcow2` image should be selected for the hard drive for the virtual machine. `qcow2` images can be converted to different virtualization formats such as VirtualBox or VMware using a tool provided by Univention, see *Providing an image for VMware/VirtualBox* (page 5).

The following settings are configured for the basic image:

- The installation language can be selected as required. The locale of the system is set based on the selected language. If you want to be able to use the appliance in more than one language, you can add another locale at a later point in time.
- A preselection is made for the time zone which is then adapted subsequently by the users of the appliance.
- The keyboard layout is only relevant for local logins; it is not important for the web-based configuration.
- A configuration via DHCP is the most practical presetting for appliance images. The Univention Installer attempts to perform a DHCP request in the scope of the network configuration. The network configuration is only performed via DHCP if this is successful, i.e., an IP address must be assigned to the appliance for the duration of the setup. This can be done with an *IP client* object in the Univention Management Console.
- In the next step, the initial password is set for the root user. This root password is changed by the end user during the commissioning of the appliance image.
- The partitioning can be performed as required, e.g., by using a LVM. For an image that will be used in a cloud setup, a single root partition should be used. This allows growing the root partition based on the selected instance disk size.

Once the basic installation is complete, a dialogue is shown in which you can select whether to create a new UCS domain or join an existing domain. To create the appliance, `Control+Q` must be pressed at this point to interrupt the process. The installation continues for a short period of time, during which the *Starting Univention System Setup* message appears and the systems then restarts.

The installation of the basic image is now complete. Following a reboot, the user of the appliance is shown the dialogue for adjusting the configuration, see *Using a UCS appliance* (page 1).

In most cases, the appliance needs to be preconfigured with a certain selection of software. The installation is usually performed via the Univention App Center, which, however, is not yet available at this point in time. The installation is thus performed via the command line. UCS standard components can be installed using the corresponding package names, e.g.

```
$ univention-install univention-printserver
```

Packages from the Univention App Center are installed with the command `univention-app install` once a valid license is available. The ID of an application can be retrieved with the command `univention-app list`:

```
$ univention-app install APPID
```

The system now needs to be shut down cleanly without file systems still being mounted.

The `qcow2` image (i.e., the hard drive of the virtual machine) is now copied. If the `default` storage pool of `libvirt` was used, the image is stored in the directory `/var/lib/libvirt/images/`.

Additional steps are required if the image is to be used in Amazon EC2 (see [Providing an image for Amazon EC2](#) (page 4)), OpenStack (see [Providing an image for OpenStack](#) (page 5)) or as a VMware / VirtualBox appliance (see [Providing an image for VMware/VirtualBox](#) (page 5)).

2.1.1 Providing an image for Amazon EC2

The following adjustments need to be made for an image that is to be used in Amazon EC2.

The following Univention Configuration Registry variables can be used to generate the GRUB configuration in this format additionally. The boot loader configuration is also adapted:

```
$ DEV='/dev/xvda' GRUB='(hd0) '
$ grub-mkdevicemap ||
  echo "${GRUB} ${DEV}" >/boot/grub/device.map
$ append="$ (ucr get grub/append |
  sed -re "s|/dev/sda|${DEV}|g;s|(no)?splash||g") "
$ xargs -d'\n' ucr set <<__EOT__
grub/append=${append}
grub/boot=${DEV}
grub/root=${DEV}1
grub/bootsplash=no
grub/quiet=no
grub/rootdelay=0
grub/timeout=0
grub/terminal=console serial
grub/serialcommand=serial --unit=0 --speed=115200 --word=8 --parity=no --stop=1
__EOT__
$ update-initramfs -uk all
$ update-grub
```

The initial login to the EC2 instance is performed via SSH keys. The temporary password for the user `root` should be removed to prevent logins. The following Univention Configuration Registry variable configures this start mode:

```
$ usermod -p \* root
$ ucr set server/amazon=true
```

The name server should be set; in this example to `OpenDNS`. Additionally, the timeout when waiting for a DHCP request answer is lowered.

```
$ ucr set nameserver1=208.67.222.222 dns/forwarder1=208.67.222.222
$ ucr unset nameserver2 nameserver3
$ ucr unset dns/forwarder2 dns/forwarder3
$ ucr set interfaces/eth0/type=dhcp dhclient/options/timeout=12
$ ucr set timeserver=169.254.169.123 # AWS internal
$ ucr set system/setup/boot/start=false
```


2.1.2 Providing an image for OpenStack

The provisioning for OpenStack images occurs via Cloud-Init (see *Automatic configuration of an appliance with Cloud-Init* (page 6)). Cloud-Init is a standardized solution for configuration of an image. Cloud-Init checks a range of data sources for an existing configuration. The package `cloud-init` must be installed to prepare an image for provisioning via Cloud-Init:

```
$ univention-install cloud-init
```

Univention System Setup should be disabled if interactive provisioning using a browser running inside the OpenStack instance is not required:

```
$ ucr set system/setup/boot/start=false
```

The initial login to the OpenStack instance is performed via SSH keys. The temporary password for the user root should be removed to prevent logins.

```
$ usermod -p \* root
```

2.1.3 Providing an image for VMware/VirtualBox

Virtualization images for **VirtualBox**, **VMware Player** and **VMware ESXi** can also be created on the basis of the `qcow2` images above. The package `generate-appliance` provides tools for this.

The `generate_appliance` tool must be started and the `qcow2` image selected with the parameter `-s`:

```
$ generate_appliance -s appliance.qcow2
```

The virtual machine is assigned one CPU and one gigabyte of RAM as standard. If the appliance requires more storage or CPUs, the parameter `-m` can be used to specify the RAM size in megabytes and `-c` can be used to assign the number of CPUs. The parameters `--vendor` and `--product` can be used to specify a vendor and product name.

By default three different virtualization images are generated from the `qcow2` image. The generation for a type can be suppressed using the respectively given option:

- Zipped VMware compatible images (e.g. for **VMware Player**), can be suppressed with `--no-vmware`
- **VirtualBox** OVA image, can be suppressed with `--no-ova-virtualbox`
- **VMware ESXi** OVA image, can be suppressed with `--no-ova-esxi`

2.2 Automatic configuration of an appliance

Instead of an interactive configuration of the appliance by the user, it can also be performed automatically. The automatic configuration can either be performed via `cloud-init` (a general tool for the provision of cloud images) or a Univention appliance mode profile file.

2.2.1 Automatic configuration with a UCS appliance mode profile file

Automatic configuration with the UCS appliance mode requires creating a profile file `/var/cache/univention-system-setup/profile`. Example configuration:

```
hostname="ucs"
domainname="testdom.example"
windows/domain="TESTDOM"
ldap/base="dc=testdom,dc=example"
root_password="univention"

locale/default="de_DE.UTF-8:UTF-8"
packages_install="univention-s4-connector univention-samba4"
packages_remove=""

server/role="domaincontroller_master"

interfaces/eth0/type="static"
interfaces/eth0/address="192.0.2.2"
interfaces/eth0/netmask="255.255.255.0"
interfaces/eth0/network="192.0.2.0"
interfaces/eth0/broadcast="192.0.2.255"
dns/forwarder1="192.0.2.1"
gateway="192.0.2.1"
```

If `interfaces/eth0/type` is set to `dhcp`, DHCP is used for the network configuration.

Then the `/usr/lib/univention-system-setup/scripts/setup-join.sh` tool needs to be run once. Then Apache and the UMC server need to be restarted:

```
$ deb-systemd-invoke restart apache2 univention-management-console-server
```

2.2.2 Automatic configuration of an appliance with Cloud-Init

Note: This chapter is not up-to-date with UCS 5.

Cloud-Init works on a configuration file in the cloud configuration format. The configuration file is provided by the respective cloud service; the type of provision differs from cloud solution to cloud solution. It is currently only possible to provide a Primary Directory Node.

The configuration file may be adapted for different scenarios. Similar to *Automatic configuration with a UCS appliance mode profile file* (page 6) a profile file must be placed at file:`/var/cache/univention-system-setup/profile` and `/usr/lib/univention-system-setup/scripts/setup-join.sh` must be invoked afterwards.

The following includes an example file with which a Primary Directory Node can be provided. In addition, several files are generated on the system: the UCS license to be installed and a file with the apps to be installed from the Univention App Center. The license in this example is the default *core edition license*. More information about requesting a proper license can be found in *License management in cloud instances* (page 8).

Two example hook scripts are generated which are called after setup is finished: One calls `wget` for a given URL, which could be used to signal an external service that the provisioning of the instance is done.

```
#cloud-config
#
write_files:
- path: /var/cache/univention-system-setup/profile
  content: |
    hostname="myucsprimary"
    domainname="ucs.example"
```

(continues on next page)

(continued from previous page)

```

windows/domain="UCS"
ldap/base="dc=ucs,dc=example"
root_password="univention"
locale/default="de_DE.UTF-8:UTF-8"
packages_install=""
packages_remove=""
server/role="domaincontroller_master"
interfaces/eth0/type="dhcp"
owner: root:root
permissions: '0644'
- path: /var/cache/univention-system-setup/license
content: |
  dn: cn=admin,cn=license,cn=univention,dc=ucs,dc=example
  objectClass: top
  objectClass: univentionLicense
  objectClass: univentionObject
  univentionObjectType: settings/license
  univentionLicenseEndDate: unlimited
  univentionLicenseModule: admin
  cn: admin
  univentionLicenseBaseDN: UCS Core Edition
  univentionLicenseUsers: unlimited
  univentionLicenseServers: unlimited
  univentionLicenseManagedClients: unlimited
  univentionLicenseCorporateClients: unlimited
  univentionLicenseVirtualDesktopUsers: 0
  univentionLicenseVirtualDesktopClients: 0
  univentionLicenseSupport: 0
  univentionLicensePremiumSupport: 0
  univentionLicenseVersion: 2
  univentionLicenseType: UCS
  univentionLicenseSignature: ZjfofoUmITUqpyF5q+Afe1i6EwsKXGWYnkh3JLJH3/
↪bXqvD26nG
  ↵
↪aLa+cpcr6g9Stkx2Lslh1feGCpsdvowkA3T+SftPHSX0Fds78QgyatoiFlA6mbbtMf3ABbMfW9Glt
  IZBbxxDFD+hMO/7yOHwaFZM3xb1I2ToJ1D2+XvOxrZe2SCZd4KJIXpupnmJnAC/
↪D4Y9iqHPytVPU3
  QlI6zXnGU5q47RN/tdXLTPv7mHoiXRWh282TNOlnEiiQxwiQ4u2ghWE1x/EWY/
↪CXvZm0PQcsFqGyB
  v72WdEUOex1Yuf3BgZ7QfL0Q2Xiv6KPKCyYqZqlSNp8Xk+IpKjDqL+aq0oyeg==
owner: root:root
permissions: '0400'
- path: /var/cache/univention-system-setup/installapps
content: |
  simplesamlphp
  adconnector
owner: root:root
permissions: '0400'
- path: /usr/lib/univention-system-setup/appliance-hooks.d/90_wget_url
content: |
  #!/bin/sh
  exec wget http://myURL/page?myparam=myValue
owner: root:root
permissions: '0755'
runcmd:
- /usr/lib/univention-system-setup/scripts/setup-join.sh

```

The file with the apps to be installed contains a list of IDs of applications from the Univention App Center, see *Performing the basic installation* (page 3). The list in the example above installs the **AD Connector** and the **SAML integration** on the provided Primary Directory Node.

2.2.3 License management in cloud instances

By default a UCS installation has a *core edition license*. An updated license from Univention is required in order to use the App Center. For standard installations it is sent to the user by email and then set up in the Univention Management Console.

Cloud service providers have the possibility of retrieving UCS licenses via an API, i.e., if a new instance is to be created for a customer, the license can be retrieved via the API and then installed in the provided instance directly.

Access to the license server requires a user name and a password. These can be requested from [Univention contact](#)¹. In this document, `https://license.univention.de/shop/example/` is used as an example URL for the license server.

API for retrieving UCS licenses

The licenses are retrieved via HTTPS from the Univention license server `license.univention.de`. The retrieval can be performed completely with **wget**.

Firstly, a session with the license server must be opened, in this case with the user name `univention` and the password `secret` as an example. It is also possible to request more than one license in one session.

```
$ wget \  
  --keep-session-cookies \  
  --save-cookies cookie.db \  
  --load-cookies cookie.db \  
  --post-data='username=univention&password=secret' \  
  https://license.univention.de/shop/example/
```

A license can also be ordered with a POST request via **wget**.

Note: Special characters such as blank spaces must be escaped in URL-encoded syntax, see <https://en.wikipedia.org/wiki/Percent-encoding> for details.

```
$ wget \  
  --keep-session-cookies \  
  --save-cookies cookie.db \  
  --load-cookies cookie.db \  
  --post-data='kundeEmail=customer@example&\  
'kundeUnternehmen=New%20Customern&\  
'EndDate=31.12.2023&\  
'BaseDN=dc%3Ddrei%2Cdc%3Dzwei%2Cdc%3Dtest&\  
'Servers=0&\  
'Support=0&\  
'PremiumSupport=0&\  
'Users=100&\  
'ManagedClients=0&\  
'CorporateClients=0&\  
'VirtualDesktopUsers=0&\  
'VirtualDesktopClients=0&\  
'Type=UCS' \  
  https://license.univention.de/shop/example/order
```

If the order is successful, the HTTP status code 202 is returned. The HTML data includes the tag `orderid`, which identifies the order number of a successful order:

```
...  
<span id="orderid">21</span>  
...
```

¹ <https://www.univention.com/contact/>

If the order fails, a HTTP status code 4xx is returned and the `details` tag includes additional information, e.g.:

```
...  
<span id="details">Not a valid date: u'31.12.2023</span>  
...
```

Should it not be possible to process an order due to a server error, 5xx is output as the return code. The order can then be repeated at a later point in time.

Following ordering of a license, it takes a few seconds before the license is generated. It can then be retrieved in LDIF format using the order number. If the request above returns e.g. the order number 465, the file name is thus `465.ldif`. The request specified below waits for the availability of the license for up to sixty seconds:

```
$ wget \
  --keep-session-cookies \
  --save-cookies cookie.db \
  --load-cookies cookie.db \
  https://license.univention.de/shop/example/orders/465.ldif
```


PROFILE-BASED INSTALLATION

In addition to the interactive installation described in the *Univention Corporate Server - Manual for users and administrators* [1], a profile-based installation of UCS is also possible. With this method, the settings for the Debian Installer and Univention System Setup are specified in a preseed file.

The Debian Installer consists of a number of special-purpose components to perform each installation task. Each component performs its task, asking the user questions as necessary to do its job. The questions themselves are given priorities, and the priority of questions to be asked is set when the installer is started.

When a default installation is performed, only essential (`priority=high`) questions will be asked. This results in a highly automated installation process with little user interaction.

If there is a problem, the user will see an error screen, and the installer menu may be shown in order to select some alternative action. Serious error notifications are set to `priority=critical` so the user will always be notified.

Power users may be more comfortable with a menu-driven interface, where each step is controlled by the user rather than the installer performing each step automatically in sequence. To use the installer in a manual, menu-driven way, add the boot argument `priority=medium`.

If your hardware requires you to pass options to kernel modules as they are installed, you will need to start the installer in “expert” mode. This can be done by adding the boot argument `priority=low`.

Depending on the selected priority the installer will ask more or less questions. The installer will either use internal default values or the values from the profile. To perform the installation fully unattended all required answers must be provided through the installation profile. Therefore `priority=critical` should be specified under *additional start options* for UCS systems using the profile from *Example installation profile* (page 12).

3.1 Structure of profile files

An installation profile is a text file which can be edited with any editor. The file must use the UTF-8 character encoding. Empty lines and lines starting with a hash character (#) are ignored. All other lines should follow the four column layout required by `debconf`, which is fully described in *Debian Installer* [2]:

```
# Comment
<owner> <question name> <question type> <value>
```

The owner of most questions will be `d-i`, which is the *Debian Installer*. The *question type* depends on the questions and can be `boolean`, `string` or `select`. Any questions not answered by the preseed file is asked interactively and will prevent an unattended installation.

3.2 Example installation profile

A template file is provided as `/usr/share/doc/univention-net-installer/examples/TEMPLATE`.

Note: The file may be stored compressed with the `.gz` extension. Use **gunzip** to extract a copy of that file.

It contains the minimum required settings to perform a fully automatic installation of a Managed Node with no additional software. It will use the German keyboard layout and language settings. It will re-partition the hard-disk without asking any questions and will use LVM to manage the disk space. No additional software will be installed.

```

1 #
2 # This file overwrites /proc/cmdline overwrites preseed.cfg in the InitRamFs!
3 #
4
5 #
6 # The following options must be set through the PXE configuration
7 #
8 # Delay asking for locale and keyboard layout after preseeding via network
9 #d-i auto-install/enable boolean true
10 # Only ask for critical questions
11 #d-i debconf/priority select critical
12 # Disable graphical installer
13 #d-i debian-installer/framebuffer boolean false
14
15 # no live installer
16 d-i live-installer/enable boolean false
17
18 #
19 # Use interfaces with link
20 #
21 d-i netcfg/dhcp_timeout string 60
22
23 #
24 # Use dummy hostname and domain
25 #
26 d-i netcfg/get_hostname string unassigned-hostname
27 d-i netcfg/get_domain string unassigned-domain
28 krb5-config krb5-config/default_realm string UNASSIGNED-REALM
29 krb5-config krb5-config/kerberos_servers string localhost
30 krb5-config krb5-config/admin_server string localhost
31
32 #
33 # Select German as default locale and for keyboard layout
34 #
35 d-i debian-installer/locale string de_DE.UTF-8
36 d-i keyboard-configuration/xkb-keymap select de(nodeadkeys)
37 #d-i keyboard-configuration/modelcode string pc105
38 d-i ucr/xorg/keyboard/options/XkbModel string pc105
39 #d-i keyboard-configuration/layoutcode string de
40 d-i ucr/xorg/keyboard/options/XkbLayout string de
41 #d-i keyboard-configuration/variantcode string nodeadkeys
42 d-i ucr/xorg/keyboard/options/XkbVariant string nodeadkeys
43 #d-i keyboard-configuration/optionscode string
44 d-i ucr/xorg/keyboard/options/XkbOptions string
45 #d-i debian-installer/keymap select de-latin1-nodeadkeys
46
47 #
48 # Configure local repository server
49 #

```

(continues on next page)

(continued from previous page)

```

50 d-i debian-installer/allow_unauthenticated boolean true
51 d-i mirror/country string manual
52 d-i mirror/protocol select http
53 d-i mirror/http/proxy string
54 # The host name of the repository server is filled through the PXE configuration.
   ↳ generated by UDM
55 #d-i mirror/http/hostname string updates.software-univention.de
56 d-i mirror/http/directory string /univention-repository/
57 d-i mirror/codename string ucs505
58 d-i mirror/suite string uc504
59 d-i mirror/udeb/suite string ucs505
60
61 #
62 # Disable password for user 'root'
63 #
64 d-i passwd/root-login boolean true
65 # Alternative: printf "secret" | mkpasswd -s -m sha-512
66 d-i passwd/root-password-crypted string *
67 d-i passwd/make-user boolean false
68
69 #
70 # Partition hard disk: Use "lvm" and one big "/" partition
71 #
72 # Choices: lvm crypto regular
73 d-i partman-auto/method string lvm
74 # Choices: atomic home multi
75 d-i partman-auto/choose_recipe string atomic
76 d-i partman-auto/init_automatically_partition select 60some_device_lvm
77 d-i partman-auto/init_automatically_partition seen false
78 d-i partman-auto-lvm/new_vg_name string vg_ucs
79 d-i partman-lvm/device_remove_lvm boolean true
80 d-i partman-md/device_remove_md boolean true
81 d-i partman-lvm/confirm boolean true
82 d-i partman-lvm/confirm_nooverwrite boolean true
83 d-i partman-partitioning/confirm_write_new_label boolean true
84 d-i partman/choose_partition select finish
85 d-i partman/confirm boolean true
86 d-i partman/confirm_nooverwrite boolean true
87
88 # Pre-select the standard UCS kernel
89 #d-i base-installer/kernel/image string linux-image-amd64
90 d-i base-installer/includes string less univention-config
91 d-i base-installer/debootstrap_script string /usr/share/debootstrap/scripts/sid
92
93 #
94 # Only minimal install
95 #
96 d-i apt-setup/use_mirror boolean false
97 d-i apt-setup/no_mirror boolean true
98 d-i apt-setup/services-select multiselect none
99 d-i apt-setup/cdrom/set-first boolean false
100 tasksel tasksel/first multiselect none
101 d-i pkgsel/include string univention-system-setup-boot univention-management-
   ↳ console-web-server univention-management-console-module-setup linux-image-amd64
   ↳ openssh-server univention-base-packages
102 postfix postfix/main_mailer_type string No configuration
103 openssh-server ssh/disable_cr_auth boolean false
104 d-i ucf/changeprompt select keep_current
105 d-i pkgsel/upgrade select none
106 popularity-contest popularity-contest/participate boolean false
107

```

(continues on next page)

(continued from previous page)

```
108 #
109 # Install GRUB in MBR by default on new systems
110 #
111 d-i grub-installer/only_debian boolean true
112 d-i grub-installer/bootdev string default
113 grub-pc grub-pc/install_devices multiselect
114 grub-pc grub-pc/install_devices_empty boolean true
115
116 #
117 # After installation
118 #
119 d-i finish-install/reboot_in_progress note
120 d-i cdrom-detect/eject boolean true
121
122 #
123 # Disable starting "Univention System Setup Boot"
124 #
125 d-i ucr/system/setup/boot/start string false
126
127 #
128 # Univention System Setup profile
129 #
130 #univention-system-setup-boot uss/root_password string
131 univention-system-setup-boot uss/components string
132 univention-system-setup-boot uss/packages_install string
133 univention-system-setup-boot uss/packages_remove string
134 # Choices: domaincontroller_master domaincontroller_backup domaincontroller_slave_
135 ↔memberserver
136 univention-system-setup-boot uss/server/role string memberserver
137 #univention-system-setup-boot uss/ldap/base string dc=example,dc=com
```

Explain example

1. Line 8: These settings must be configured as PXE command line parameters in *additional start options*. They are listed here for reference only and cannot be changed through this file:

- The parameter `auto-install/enable` is used to switch the order of some installer modules: The network should be configured and the `preseed.cfg` should be loaded *before* the first questions about the locale settings are asked.
- The parameter `netcfg/choose_interface=auto` tells the installer to use the same interface which was used for the PXE boot.
- Also some of those early questions are asked at priority level `high`. The priority level should be raised to `critical` to hide them.

The long parameter names can be abbreviated as `auto=true priority=critical interface=auto`.

2. Line 33: If the locale settings are not consistent, the installer will ask interactively for corrections. The keyboard related settings must be configured through Univention Configuration Registry - the questions starting with `keyboard-configuration/xkb-...` will not work!
3. Line 55: The location of the local repository is filled in through the PXE configuration. By default the value of the Univention Configuration Registry Variable `repository/online/server` is used. It can be over-written by specifying the value here in the profile file. For use with the public repository specify `updates.software-univention.de` here.
4. Line 66: By default no password is set, which will prevent logging in. It should be replaced by an encrypted password, which can be used by running a command like `printf "secret" | mkpasswd -s -m sha-512`

5. Line 70: By default all existing partitions will be wiped without asking any question! They will be replaced by a single file system for / using LVM. See *Debian Installer* [2] for more advanced partitioning schemas.
6. Line 123: This section contains the UCS specific settings, which are normally configured through Univention System Setup. For an unattended installation the graphical installer is disabled. All other values starting with `uss/` are copied to the installation profile. The variables are described in *Overview of profile variables* (page 15).

3.3 Overview of profile variables

3.3.1 Profile variables - System properties

The following profile variables can be used to specify basic properties of the computer such as the computer name, its role within the UCS domain and the name of the domain the computer should join.

Table 3.1: Profile variables - System properties

Name	Function
<code>server/role</code>	The system role. You may choose from <code>domaincontroller_master</code> (for Primary Directory Node), <code>domaincontroller_backup</code> (for Backup Directory Node), <code>domaincontroller_slave</code> (for Replica Directory Node) and <code>memberserver</code> (for Managed Node). The properties of the system roles are described in the domain services chapter of the <i>Univention Corporate Server - Manual for users and administrators</i> [1].
<code>hostname</code>	The computer name. The name must only contain the letters a to z in lowercase, the figures 0 to 9 and hyphens. Although underscore are allowed as well, they should not be used as they are not supported everywhere. The name must begin with a letter.
<code>domainname</code>	The name of the DNS domain in which the computer is joined.
<code>windows/domain</code>	The name of the NetBIOS domain used by Samba. This variable should only be defined for the system role Primary Directory Node.
<code>locales</code>	Localization packages to be installed (locales). If more than one locale is specified, the locales are separated by blank spaces.
<code>locale/default</code>	The standard locale for the computer, e.g. <code>en_GB.UTF-8:UTF-8</code> . More information on system locales can be found at <i>Locale - Debian Wiki</i> [3].
<code>country, keymap</code>	The keyboard layout for the computer, specified in the form of an X11 key map entry, e.g. <code>de-latin1</code> .
<code>timezone</code>	The time zone for the computer, e.g. <code>Europe/Berlin</code> . A complete list of possible configuration options is shown in the <i>Basic settings</i> module of the Univention Management Console.
<code>root_password</code>	The password for the <code>root</code> user for this computer. On a Primary Directory Node, this password is also used for the <code>Administrator</code> password.

3.3.2 Profile variables - LDAP settings and domain joins

Automatically joining the computer into the domain is currently not supported for security reasons.

Table 3.2: Profile variables - LDAP settings and domain joins

Name	Function
<code>start/join</code>	As standard, all computers apart from the Primary Directory Node attempt to join the UCS domain in the course of the installation. If this parameter is set to <code>false</code> , the automatic domain join is deactivated.
<code>ldap/base</code>	The base DN of the LDAP domain. In general, the base DN <code>dc=example, dc=com</code> is used in a domain <code>example.com</code> . This variable is only evaluated on the system role Primary Directory Node.

3.3.3 Profile variables - Network configuration

By default automatically installed systems use DHCP. The following profile variables can be used to specify the network configuration of the computer.

General information on the network configuration and the use of the name servers can be found in Chapter *Network configuration* of the *Univention Corporate Server - Manual for users and administrators* [1].

The settings for network cards must be performed completely. It is not possible to leave individual settings blank. For example, if there is no IP address for the device `eth0` in the profile, in addition to the IP address, the `interfaces/eth0/netmask` will also be requested.

Table 3.3: Profile variables - Network configuration

Name	Function
<code>interfaces/ethN/type</code>	If this parameter is set to <code>dynamic</code> or <code>dhcp</code> , the network interface <code>ethN</code> procures its network configuration via DHCP. The settings of <code>interfaces/ethN/address</code> , <code>interfaces/ethN/netmask</code> , <code>interfaces/ethN/network</code> , <code>interfaces/ethN/broadcast</code> , <code>nameserverN</code> and <code>gateway</code> then become optional, but can still be used to over-write the configuration provided by DHCP. If no DHCP offer is received, a random IP address from the link-local network <code>169.254.x.x</code> is used. For manual configuration this parameter must be set to <code>static</code> .
<code>interfaces/ethN/address</code>	The IPv4 address of the physical network interface <code>ethN</code> .
<code>interfaces/ethN/netmask</code>	The network mask of the subnetwork from which the IPv4 address of <code>ethN</code> originates.
<code>gateway</code>	The IPv4 address of the gateway which the computer should use as standard. Alternatively, one can specify the computer name or the FQDN that can be resolved into the IP address.
<code>interfaces/ethN/ipv6/name/address</code>	An IPv6 address of the physical network interface <code>ethN</code> in static configuration. Multiple addresses can be assigned by using different name prefixes.
<code>interfaces/ethN/ipv6/name/prefix</code>	The prefix length of the IPv6 address of the physical network interface <code>ethN</code> in static configuration.
<code>ipv6/gateway</code>	The IPv6 address of the gateway which the computer should use as standard. It is not obligatory to enter a gateway for IPv6, but recommended. An IPv6 gateway configured here has preference over router advertisements, which might otherwise be able to change the route.
<code>interfaces/ethN/acceptRA</code>	If this setting is set to <code>yes</code> , the stateless address auto-configuration (SLAAC) is used. In this case, the IP address is assigned from the routers of the local network segment. If the variable is set to <code>no</code> , the configuration is performed statically via <code>interfaces/ethN/ip6</code> and <code>interfaces/ethN/prefix6</code> (see there).
<code>nameserver1, nameserver2, nameserver3</code>	The IP address of the name server which should perform the name resolution. It is possible to specify up to three name servers.
<code>dns/forwarder1, dns/forwarder2, dns/forwarder3</code>	The IP address of the name server intended to serve as the forwarder for a locally installed DNS service. It is possible to specify up to three forwarders.
<code>proxy/http</code>	The URL of a proxy server to be used when accessing the internet. The specified URL is adopted in the Univention Configuration Registry variables <code>proxy/http</code> and <code>proxy/ftp</code> . This setting is only required if packages are to be installed which download additional packages from external web servers; e.g., the installation program for the Flash plugin. Example: <code>proxy/http="http://proxy.example.com:8080"</code>

3.3.4 Profile variables - Software selection

The following profile variables refer to software packages which are to be installed on the computer.

Table 3.4: Profile variables - Software selection

Name	Function
<code>packages_install</code>	This settings names packages which are additionally installed. If more than one package is specified, the packages are separated by blank spaces.
<code>packages_remove</code>	This settings names packages which should be removed. If more than one package is specified, the packages are separated by blank spaces.

3.3.5 Profile variables - SSL

A SSL certification infrastructure is set up during installation of a Primary Directory Node. If no settings are configured, automatic names are given for the certificate.

Table 3.5: Profile variables - SSL

Name	Function
<code>ssl/country</code>	The ISO country code of the certification body appearing in the certificate (root CA), specified with two capital letters.
<code>ssl/state</code>	The region, county or province that appears in the certificate of the root CA.
<code>ssl/locality</code>	Place appearing in the certificate of the root CA.
<code>ssl/organization</code>	Name of the organization that appears in the certificate of the root CA.
<code>ssl/organizationalunit</code>	Name of the organizational unit or department of the organization that appears in the certificate of the root CA.
<code>ssl/email</code>	Email address that appears in the certificate of the root CA.

3.4 Network-based PXE installations with Univention Net Installer

Network-based, profile-based installations via PXE are performed with the Univention Net Installer, which can be set up using the package `univention-net-installer`. This installs the required TFTP server and WWW server configuration. In addition a DHCP server is required, which is provided by the package `univention-dhcp`. If the DHCP server and the PXE server of the Univention Net Installer are operated on separate systems, the PXE server must be assigned via a DHCP boot policy.

```
$ univention-install univention-net-installer univention-dhcp
```

The installation process consists of multiple steps, which contact different services and servers:

1. First the *DHCP server* is contacted. It sends the client to the *Boot server* (by default the DHCP server itself) configured through the *DHCP Boot* policy to request the boot loader given in *Boot filename* (`pxelinux.0`).
2. Then the client downloads the boot loader via the TFTP protocol from the *PXE server*. The boot loader scans the server for the client configuration file in `pxelinux.cfg/`. The referenced Linux kernel (`linux`) and initial RAM disk file (`initrd.gz`) are then downloaded. Those names can be changed through the Univention Configuration Registry Variables `pxe/installer/kernel` and `pxe/installer/initrd`.

Note: Newer versions of the PXE boot loader support downloading through `http`, which can be faster and more reliable in certain environments. This can be enabled by specifying URLs starting with `http://` as file names.

3. Finally the UCS installer downloads the profiles and package files using `http`. The *Name of the installation profile* is configured in the computer entry in LDAP. The file is fetched from the *PXE server* by default, but

the prefix can be overwritten through the Univention Configuration Registry Variable `pxe/installer/profiles`. As an alternative the name can also be specified as an absolute URL.

4. The package files are fetched from the *repository server*, which is configured through the Univention Configuration Registry Variable `repository/online/server` on the PXE server.

Univention Net Installer supports both the interactive and profile-based installation. Any questions not answered in the preseed file forces the installer to interactive mode.

Profiles should be copied into the directory `/var/lib/univention-client-boot/preseed/` on the PXE server, which is accessible through `http://HOST-NAME/univention-client-boot/preseed/`.

Univention Net Installer can either directly use the repository server <https://updates.software-univention.de/> or a local repository server. The later one is advisable as it reduces the amount of data needing to be downloaded for each installation.

3.4.1 Local repository

The local repository must first be initialized once using the command `univention-repository-create`. Since UCS 5.0-0 the PXE kernel and installer must be copied manually from the ISO image to the correct location in `/var/lib/univention-client-boot/installer/`.

```
$ mount /dev/cdrom /media/cdrom0
$ install -m644 /media/cdrom0/netboot/linux \
  /var/lib/univention-client-boot/
$ install -m644 /media/cdrom0/netboot/initrd.gz \
  /var/lib/univention-client-boot/
$ umount /media/cdrom0
```

Instead of mounting the DVD a downloaded ISO image can also be mounted by using `mount -o loop, ro /path/to/UCS.iso /media/cdrom0`. Alternatively the files can be downloaded from <http://updates.software-univention.de/pxe/5.0-5/amd64/gtk/debian-installer/amd64/>:

```
$ cd /var/lib/univention-client-boot/
$ PXE='http://updates.software-univention.de/pxe/'
$ PXE+=$(ucr filter <<<'@%@version/version@%-@%@version/patchlevel@%@')
$ PXE+=/amd64/gtk/debian-installer/amd64
$ wget -O linux "$PXE/linux"
$ wget -O initrd.gz "$PXE/initrd.gz"
```

The procedure should be repeated for each new release. Otherwise new installations will still start with an older release, which might require extra time for updating. For more information on local repositories, see the software deployment chapter of the *Univention Corporate Server - Manual for users and administrators* [1].

3.4.2 Public repository

Even when the public repository server <https://updates.software-univention.de/> is used, some services and files must be available inside the local network. At minimum this includes the DHCP service, which assigns the client its IP address and tells it to continue fetching files from the next server. Historically this had to be a TFTP server, but nowadays this also can be any HTTP server. This has the benefit that HTTP is faster, more reliable and also works over the internet.

1. Install the HTTP capable boot loader `lpxelinux.0`

```
$ ln -s /usr/lib/PXELINUX/lpxelinux.0 \
  /var/lib/univention-client-boot/
```

1. Setup the *DHCP Boot* policy to use `lpxelinux.0`. Depending on the capabilities of the network card boot code the boot loader can either be fetched over the HTTP or TFTP protocol:
 - a. For HTTP configure the absolute URL as the boot filename:

```

$ HOST="$(hostname -f) "
$ LDAP="$(ucr get ldap/base) "
$ HTTP="http://$HOST/univention-client-boot/lpxelinux.0"
$ udm policies/dhcp_boot modify \
  --dn "cn=default-settings,cn=boot,cn=dhcp,cn=policies,$LDAP" \
  --set boot_filename="$HTTP" \
  --set boot_server=

```

- b. The installer performs its own second DHCP request. This again retrieves the DHCP option `boot filename`, which now contains the *URL* to the PXE loader. The installer wrongly interprets this as the URL for the profile `preseed`, which breaks the installation. Therefore the option needs to be overwritten when the installer performs this second query:

```

$ STMT='if substring (option vendor-class-identifier, 0, 3) = "d-i" {
↳ filename ""; }'
$ udm dhcp/subnet list | \
  sed -ne 's/^DN: //p' | \
  xargs -d '\n' -n1 udm dhcp/subnet modify \
  --option options \
  --append statements="$STMT" \
  --dn

```

- For TFTP change `boot filename` to point to `lpxelinux.0`:

```

$ HOST="$(hostname -f) "
$ LDAP="$(ucr get ldap/base) "
$ udm policies/dhcp_boot modify \
  --dn "cn=default-settings,cn=boot,cn=dhcp,cn=policies,$LDAP" \
  --set boot_filename='lpxelinux.0' \
  --set boot_server="$HOST"

```

2. Configure the boot loader to load the Linux kernel and initial ram disk from the public repository server:

```

$ PXE='http://updates.software-univention.de/pxe'
$ PXE="$PXE/5.0-5/amd64/gtk/debian-installer/amd64"
$ ucr set \
  pxe/installer/kernel="$PXE/linux" \
  pxe/installer/initrd="$PXE/initrd.gz" \
  pxe/installer/ipappend=3

```

3. In the profile file the settings for `mirror/http/hostname` and `mirror/http/directory` must be changed to use the public server and its layout:

```

d-i mirror/http/hostname string updates.software-univention.de
d-i mirror/http/directory string /

```

3.4.3 Assignment of a computer for automatic installation

A computer to be installed via Univention Net Installer must firstly be registered in the computer management of the Univention Management Console. The following values must be set as a minimum at the *General* tab:

- Hostname
- MAC address
- IP address
- DNS forward and reverse zone entries
- DHCP service entry

The *(Re-)install on next boot* option must now be activated in the *Advanced settings* tab under *Deployment*.

The name of the installation profile relative to `/var/lib/univention-client-boot/preseed/` can be entered under *Name of installation profile*. As an alternative any other http server can be used as well, in which case an absolute URL must be given.

Options entered under *additional start options* are passed on to the kernel in network-based installations, e.g., for the deactivation of ACPI during system start. This can also be used to specify other preseed variables on a host-by-host basis. To perform an installation fully unattended see the *Explain example* (page 14) for a list of required options.

A PXE configuration file is created for every computer object under `/var/lib/univention-client-boot/pxelinux.cfg/`.

Tip: Several Univention Configuration Registry Variable exist on the PXE server, which can be used to further customize the PXE configuration. Use `ucr search ^pxe/` to get a list of them including a short description. Those values will only be used when next a PXE configuration file is generated.

It must be verified that the boot order in BIOS of the system to be installed prefers a PXE network boot over hard disks or CD-ROMs.

On the next restart of the computer it will boot via PXE and is installed via the network.

By default the *(Re-)install on next boot* option needs to be reset manually after the installation has finished. Otherwise the computer will be reinstalled every time the host is booted! If the package `univention-net-installer-daemon` is installed on the server, the flag can be reset automatically.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] *Univention Corporate Server - Manual for users and administrators*. Univention GmbH, 2021. URL: <https://docs.software-univention.de/manual/5.0/en/>.
- [2] *Debian Installer*. Debian Project, 2009. URL: <https://www.debian.org/releases/buster/amd64/apb.en.html>.
- [3] *Locale - Debian Wiki*. Debian Project, 2013. URL: <https://wiki.debian.org/Locale>.

INDEX

E

environment variable
 interfaces/eth0/type, 6
 proxy/ftp, 16
 proxy/http, 16
 pxe/installer/initrd, 17
 pxe/installer/kernel, 17
 pxe/installer/profiles, 18
 repository/online/server, 14, 18

I

interfaces/eth0/type, 6

P

proxy/ftp, 16
proxy/http, 16
pxe/installer/initrd, 17
pxe/installer/kernel, 17
pxe/installer/profiles, 18

R

repository/online/server, 14, 18