



# UCS 5.0 for Debian and Ubuntu Administrators

*Release 5.0-7*

Univention GmbH

Mar 14, 2024

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

## CONTENTS:

<b>1</b>	<b>System configuration</b>	<b>3</b>
1.1	Template system for configuration files . . . . .	3
1.2	Listener modules writing configuration files . . . . .	4
1.3	Consequences of manually editing configuration files . . . . .	5
1.4	Customize configuration . . . . .	5
<b>2</b>	<b>Software management</b>	<b>7</b>
2.1	Default software on UCS . . . . .	7
2.2	Software repositories and packages . . . . .	7
2.3	Commands for software management . . . . .	8
2.4	Automatic software updates . . . . .	9
2.5	Beyond software packages . . . . .	9
2.6	Third party sources . . . . .	9
<b>3</b>	<b>Domain join</b>	<b>11</b>
3.1	Join UCS systems . . . . .	11
3.2	Join scripts . . . . .	12
3.3	Consequences of unfinished join scripts . . . . .	12
<b>4</b>	<b>Principles</b>	<b>13</b>
	<b>Bibliography</b>	<b>15</b>
	<b>Index</b>	<b>17</b>



Welcome to the manual *Univention Corporate Server for Debian and Ubuntu Administrators*. This document aims for system administrators, system engineers, or DevOps engineers with knowledge of the administration of Debian GNU/Linux or Ubuntu systems who want to get started with UCS (Univention Corporate Server).

The document helps you avoid common pitfalls when starting to work with UCS if you have a Debian GNU/Linux or Ubuntu background. The document doesn't describe in detail how to do various tasks *the UCS way*. Instead, it points out the differences and refers to appropriate resources in the product documentation.

Reading this document, you earn a general understanding of the differences between Debian GNU/Linux or Ubuntu and UCS. A *list of principles* (page 13) summarizes the key points of this document.

This document assumes that you are familiar with the following topics:

- Installing and updating computer systems running Debian GNU/Linux or Ubuntu.
- A command line in a local or remote shell, such as **bash**.
- A text editor on the command line, such as **GNU nano**, **Vim**, or **Emacs**.
- Installing and upgrading software using the package manager **APT**.
- Configuring various services such as mail servers, LDAP servers and Samba.

Your feedback is welcome and highly appreciated. If you have any comments, suggestions, or criticisms, please [send your feedback](#)<sup>1</sup> to improve the document.

---

<sup>1</sup> <https://www.univention.com/feedback/?debian-admins=generic>



## SYSTEM CONFIGURATION

This section covers the differences of UCS in the area of system configuration. The term *system configuration* in the context of this section refers to controlling the behavior of a system and its services through configuration files.

Administrators typically configure Linux operating systems using text-based configuration files. Debian GNU/Linux and Ubuntu are no exception to this principle. The advantage of a text-based configuration format is that it requires no extra tools or knowledge other than a text editor and a documentation of the configuration syntax. Text-based configuration is also useful for configuration management and infrastructure as code software such as **Ansible**, **Puppet**, or **Terraform** that allow administrators to configure multiple systems.

UCS, being a derivative of Debian GNU/Linux and thus a Linux operating system, is no exception to this principle. The services on a UCS system use their familiar text-based configuration files. As an administrator with Linux know-how, it's natural to manually edit the configuration with a text editor and adapt the settings of a service to the needs of their environment.

### 1.1 Template system for configuration files

When it comes to manually editing configuration files, UCS is different from other Linux distributions. UCS automatically generates most configuration files from text-based file templates and a key-value-based variable store. Univention calls this capability *Univention Configuration Registry (UCR)*.

The registry contains various variables and their values, for example the hostname of the central LDAP server. Configuration files that need this particular value, refer to the variable. Changes to the variable trigger the UCR mechanism to recreate all affected configuration files.

#### See also:

For a detailed elaboration, see the following sections in *Univention Corporate Server 5.0 Architecture* [1]:

- [Configuration management](#)<sup>2</sup>
- [Univention Configuration Registry \(UCR\)](#)<sup>3</sup>

UCS uses UCR (Univention Configuration Registry) for the following non exhaustive list of benefits:

- Single setting values can apply to multiple configuration files.
- UCR ensures configuration consistency for services affected on a system.
- The majority of software packages on UCS use UCR for their configuration.
- Avoid configuration errors resulting from an invalid configuration setting syntax.
- Apply logic to values and write the result in the configuration file.
- Logging of changes to UCR variables.
- Applying settings to configuration files of upgraded software.

---

<sup>2</sup> <https://docs.software-univention.de/architecture/5.0/en/components/management-system.html#component-configuration-registry>

<sup>3</sup> <https://docs.software-univention.de/architecture/5.0/en/services/ucr.html#services-ucr>

UCR does not restart a service. Operators must do that manually.

You can recognize configuration files controlled by UCR by the presence of a header at the beginning of the file, as shown in Listing 1.1. The header lists the template file used to generate the configuration file:

Listing 1.1: Header in configuration files under control of UCR

```
# Warning: This file is auto-generated and might be overwritten by
#         univention-config-registry.
#         Please edit the following file(s) instead:
```

### See also:

For detailed information about how to use UCR, see the following sections in *Univention Corporate Server - Manual for users and administrators* [2]:

- Administration of local system configuration with Univention Configuration Registry<sup>4</sup>
- Using the command line frontend<sup>5</sup>
- Modifying UCR templates<sup>6</sup>

## 1.2 Listener modules writing configuration files

As an operating system utilizing a Domain concept<sup>7</sup> and a Replication concept<sup>8</sup>, the so-called *Listener-/Notifier mechanism* also generates configuration files based on configuration data in the domain database and UCR.

Changes to UCR don't trigger a regeneration of those files. Changes to related objects in the domain database, so-called UDM objects, trigger a regeneration, as well as, the command to re-synchronize a listener module.

---

**Important:** Unfortunately, listener modules that generate configuration files don't add a header with a warning to configuration files. For a status, see [Bug #56790](#)<sup>9</sup>.

---

### See also:

For more information, see the following resources in *Univention Corporate Server - Manual for users and administrators* [2]:

- Listener/notifier domain replication<sup>10</sup>
- Reinitialization of listener modules<sup>11</sup>

For more information about the concepts, see the following resources in *Univention Corporate Server 5.0 Architecture* [1]:

- Domain concept<sup>12</sup>
- Replication concept<sup>13</sup>

---

<sup>4</sup> <https://docs.software-univention.de/manual/5.0/en/computers/ucr.html#computers-administration-of-local-system-configuration-with-univention-configuration>

<sup>5</sup> <https://docs.software-univention.de/manual/5.0/en/computers/ucr.html#computers-using-the-command-line-front-end>

<sup>6</sup> <https://docs.software-univention.de/manual/5.0/en/computers/ucr.html#ucr-templates-extend>

<sup>7</sup> <https://docs.software-univention.de/architecture/5.0/en/concepts/domain.html#concept-domain>

<sup>8</sup> <https://docs.software-univention.de/architecture/5.0/en/concepts/replication.html#concept-replication>

<sup>9</sup> [https://forge.univention.org/bugzilla/show\\_bug.cgi?id=56790](https://forge.univention.org/bugzilla/show_bug.cgi?id=56790)

<sup>10</sup> <https://docs.software-univention.de/manual/5.0/en/domain-ldap/listener-notifier.html#domain-listener-notifier>

<sup>11</sup> <https://docs.software-univention.de/manual/5.0/en/domain-ldap/listener-notifier.html#domain-listener-notifier-erroranalysis-reinit>

<sup>12</sup> <https://docs.software-univention.de/architecture/5.0/en/concepts/domain.html#concept-domain>

<sup>13</sup> <https://docs.software-univention.de/architecture/5.0/en/concepts/replication.html#concept-replication>



## 1.3 Consequences of manually editing configuration files

Editing configuration files manually, as you practice it for other Linux distributions, may bring you closer to your goal. At first glance, you have a good feeling, because the configuration works as expected. The feeling can turn into frustration when the carefully handmade configuration vanished. Software package updates and installation of additional software trigger UCR to regenerate configuration files and therefore overwriting custom changes. Restarting the service, or rebooting the system, activates the changes in the configuration files.

Sometimes the causes of problems are hard to find. Each run of UCR can affect the configuration, overwrite manual changes, and require additional analysis and repair.

The negative consequences range from small effects, such as a service reverting to its default behavior, to key services not running at all. For example, the LDAP server refuses to start, preventing users from signing in their environment and preventing them from doing their daily work.

This brief overview introduces the first principle. Applying it can save you a lot of frustration, effort, and headaches.

---

### Principle #1

Don't manually edit configuration files that are under control of UCR or listener modules.

---

## 1.4 Customize configuration

Nevertheless, UCS is an open system and wants to enable administrators to customize it to their needs. To avoid problems caused by ignoring UCR, use it to your advantage.

1. Take existing UCR variables and the UCR mechanism to customize a UCS system to your needs.

To add a custom configuration, not covered by existing UCR variables, use one of the following possibilities:

2. [Modifying UCR templates](#)<sup>14</sup> and customize them to your needs.

However, keep in mind that the template file is under the control of a software package. After a package update you may need to manually merge your customization with the update that Univention provides for the package. As long as the merge isn't resolved, the affected service may not work at all.

3. Use local configuration possibilities such as `local.conf` files or local configuration directories.

The local configuration possibilities strongly depend on the service you have in mind and what configuration options it offers.

Examples:

- The Apache web server
- The print service offers a local configuration, see [Setting the local configuration properties of a print server](#)<sup>15</sup>.
- The **Samba** domain services
- The IMAP server **Dovecot**
- File and print shares

### See also:

See the following resources in *Univention Corporate Server - Manual for users and administrators* [2]

- [Administration of local system configuration with Univention Configuration Registry](#)<sup>16</sup>

---

<sup>14</sup> <https://docs.software-univention.de/manual/5.0/en/computers/ucr.html#ucr-templates-extend>

<sup>15</sup> <https://docs.software-univention.de/manual/5.0/en/print-services/configuration.html#print-services-configuration>

<sup>16</sup> <https://docs.software-univention.de/manual/5.0/en/computers/ucr.html#computers-administration-of-local-system-configuration-with-univention-configuration>

- Modifying UCR templates<sup>17</sup>

---

<sup>17</sup> <https://docs.software-univention.de/manual/5.0/en/computers/ucr.html#ucr-templates-extend>

## SOFTWARE MANAGEMENT

This section describes the differences of UCS with respect to software management. In this section, the term *software management* refers to the lifecycle of software on a computer system, such as installing software on a computer system, updating, and removing existing software.

As an enterprise Linux distribution, UCS has defined *maintenance cycles*<sup>18</sup> for its version levels and uses terms such as *errata updates* or *patch level releases*.

### See also:

- *Differentiation of update variants / UCS versions*<sup>19</sup> in *Univention Corporate Server - Manual for users and administrators* [2].
- *Maintenance lifecycle of UCS*<sup>20</sup>
- *Maintenance lifecycle rules for UCS*<sup>21</sup>

## 2.1 Default software on UCS

Univention uses many open source products and adapts them to UCS, such as **OpenLDAP**, **OpenSSL**, **Samba**, **Apache HTTP Server**. The product capability relies on these products. Replacing the existing packages with the personal favorite breaks UCS. For example, using **nginx** instead of the UCS default **Apache HTTP Server**.

---

### Principle #2

Use the standard software packages that UCS installs to provide core product capabilities.

Don't replace these software packages with software of your own personal taste.

---

## 2.2 Software repositories and packages

As a Linux distribution based on Debian GNU/Linux, UCS uses software packages in the deb file format and the corresponding package repositories. Univention provides and maintains software packages in package repositories. The packages have two different maintenance statuses: *maintained* and *unmaintained*.

### See also:

- *Packages*<sup>22</sup> in *Univention Corporate Server 5.0 Architecture* [1].
- *Configuration of the repository server for updates and package installations*<sup>23</sup> in *Univention Corporate Server - Manual for users and administrators* [2].

---

<sup>18</sup> <https://docs.software-univention.de/n/en/maintenance/ucs.html#maintenance-ucs-rules>

<sup>19</sup> <https://docs.software-univention.de/manual/5.0/en/software/ucs-versions.html#computers-differentiation-of-update-variants-ucs-versions>

<sup>20</sup> <https://docs.software-univention.de/n/en/maintenance/ucs.html#maintenance-ucs>

<sup>21</sup> <https://docs.software-univention.de/n/en/maintenance/ucs.html#maintenance-ucs-rules>

<sup>22</sup> <https://docs.software-univention.de/architecture/5.0/en/positioning.html#positioning-packages>

<sup>23</sup> <https://docs.software-univention.de/manual/5.0/en/software/repository-server.html#software-config-repo>

## 2.3 Commands for software management

To install, update, and remove software packages from a Debian GNU/Linux system, the Debian Project provides the commands **apt** for interaction with the package manager and **apt-get** for non-interactive interaction. For graphical interaction, there are tools such as **aptitude** and **synaptic**. As a Debian GNU/Linux or Ubuntu administrator you are familiar with these tools.

UCS provides the following command line tools as wrappers around **apt-get**:

### **univention-install**

to install software packages from a software repository on UCS. This command also implicitly updates the package cache on a UCS system.

### **univention-upgrade**

to update installed software packages on a UCS system, for example errata updates or patch level releases.

### **univention-remove**

to remove installed software packages from UCS.

### **univention-app**

to install, update, or remove apps in the App Center on UCS.

---

### Principle #3

Use the **univention-\*** tools to perform actions for installing, updating and removing software packages and apps on UCS.

---

In contrast to **apt** and **apt-get**, the **univention-\*** commands take care of the following additional aspects of software management on UCS:

1. Not all administrators run **apt update** before installing software. **univention-install** always updates the local software package cache before installing software. This ensures that the package manager installs the latest stable software version.
2. **univention-install** ensures the transfer of configuration settings, for example from *UCR variables* (page 3) or *join scripts* (page 11).

For example, when installing **Postfix** with **apt**, the standard wizard of the package asks for the type of the mail system, such as *Internet Site*, *Internet with smarthost*, *Satellite system*, or *Local only*. **univention-install** doesn't run the wizard. Instead, it applies the relevant system configuration settings so that the mail server works after the installation and is ready for your adjustments, if required.

3. The installation of meta packages ensures that the package manager uses UCS mechanisms during the installation, such as the correct configuration of the software package and information storage in the domain.

### See also:

For further information about the mentioned commands, see the following sections in *Univention Corporate Server - Manual for users and administrators* [2]:

- [Installation/removal of individual packages in the command line](#)<sup>24</sup>
- [Univention App Center](#)<sup>25</sup>

---

<sup>24</sup> <https://docs.software-univention.de/manual/5.0/en/software/further-software.html#computers-installation-removal-of-individual-packages-in-the-command-line>

<sup>25</sup> <https://docs.software-univention.de/manual/5.0/en/software/app-center.html#software-appcenter>

## 2.4 Automatic software updates

UCS uses policies to define automatic software updates for systems in a domain.

### See also:

For more information, see the following sections in *Univention Corporate Server - Manual for users and administrators* [2]:

- [Updating systems via a policy](#)<sup>26</sup>
- [Specification of an update point using the package maintenance policy](#)<sup>27</sup>

## 2.5 Beyond software packages

In addition to the well-known software packages, Univention also distributes software as apps through *Univention App Center*. Apps provide a connector between UCS and the software. The app can be standalone or accompanied by the software itself. For example, to install apps such as UCS components, for example **Active Directory-compatible Domain Controller**, or third-party software such as **Nextcloud** or **ownCloud**, you must use the App Center, either the corresponding UMC module or the **univention-app** command. Most apps use Docker images and offer a ready-to-use integration with UCS.

### See also:

For more information, see the following resources:

- [Univention App Center](#)<sup>28</sup> in *Univention Corporate Server - Manual for users and administrators* [2]:
- [Univention app ecosystem](#)<sup>29</sup> in *Univention Corporate Server 5.0 Architecture* [1]:
- [Univention App Center Catalog](#)<sup>30</sup>

## 2.6 Third party sources

As a Debian GNU/Linux or Ubuntu administrator you know about the ability to add third-party software repositories to install additional software on your system.

You can also add third-party repositories to UCS by editing the sources lists or adding files to `/etc/apt/sources.list.d`. While this is possible, it's not recommended. Be aware of the possible negative consequences, such as breaking existing services.

For example, adding PHP packages from another PHP repository interferes with the existing PHP packages and may negatively affect other software from the UCS software repository that relies on the default PHP version in UCS. Product tests only cover software packages from the Univention software repository.

With regard to additional Python packages, don't install packages through **pip** into your system-wide Python environment, but into a virtual environment instead. See [venv - Creation of virtual environments](#)<sup>31</sup>. Installing or updating Python packages with **pip** in the system-wide Python environment of UCS, can overwrite existing files and can cause problems that are difficult to diagnose.

---

### Principle #4

Before installing third-party software packages:

<sup>26</sup> <https://docs.software-univention.de/manual/5.0/en/software/updates.html#computers-softwaremanagement-release-policy>

<sup>27</sup> <https://docs.software-univention.de/manual/5.0/en/software/package-maintenance-policy.html#computers-softwaremanagement-maintenance-policy>

<sup>28</sup> <https://docs.software-univention.de/manual/5.0/en/software/app-center.html#software-appcenter>

<sup>29</sup> <https://docs.software-univention.de/architecture/5.0/en/concepts/app-center.html#univention-app-ecosystem>

<sup>30</sup> <https://www.univention.com/products/app-catalog/>

<sup>31</sup> <https://docs.python.org/3.7/library/venv.html#module-venv>

1. Always verify the App Center and the standard Univention software repositories to see if the software is already available there.
  2. Make sure that the packages don't overwrite existing packages.
  3. Prefer Docker images to Debian packages.
  4. Use **pip** in virtual Python environments only.
-

## DOMAIN JOIN

This section covers aspects of the domain join that administrators of Debian GNU/Linux or Ubuntu systems need to be aware of regarding UCS.

UCS aims for multi-server environments. The first system takes the server role of the *Primary Directory Node* in a UCS domain. A domain is a single trust context that groups one or more entities such as computer systems or users. The domain provides domain services to systems and users.

### See also:

For more information in *Univention Corporate Server 5.0 Architecture* [1]:

- [Domain concept](#)<sup>32</sup> about the most important concept in UCS.
- [Role concept](#)<sup>33</sup> about the different roles of UCS systems in a UCS domain.
- [Permission concept](#)<sup>34</sup> about the different permissions of default user groups.

For more information in *Univention Corporate Server - Manual for users and administrators* [2]:

- [UCS system roles](#)<sup>35</sup>
- [Primary Directory Node](#)<sup>36</sup>

## 3.1 Join UCS systems

To join a UCS system to an existing UCS domain, use the possibilities outlined in [How UCS systems join domains](#)<sup>37</sup>.

---

### Principle #5

Don't run `univention-join` on a *Primary Directory Node*. It just skips.

---

<sup>32</sup> <https://docs.software-univention.de/architecture/5.0/en/concepts/domain.html#concept-domain>

<sup>33</sup> <https://docs.software-univention.de/architecture/5.0/en/concepts/role.html#concept-role>

<sup>34</sup> <https://docs.software-univention.de/architecture/5.0/en/concepts/permission.html#concept-permission>

<sup>35</sup> <https://docs.software-univention.de/manual/5.0/en/domain-ldap/system-roles.html#system-roles>

<sup>36</sup> <https://docs.software-univention.de/manual/5.0/en/domain-ldap/system-roles.html#domain-ldap-primary-directory-node>

<sup>37</sup> <https://docs.software-univention.de/manual/5.0/en/domain-ldap/domain-join.html#linux-domain-join>

## 3.2 Join scripts

Services and apps that integrate with the domain, provide so-called *join scripts*. A service's join script requires the credentials of a domain administrator to write data to the domain database so that the administrator can manage it.

Installing UCS components through the App Center ensures that the join scripts run after the installation. If administrators install the same component using the package manager, the join scripts don't run and the administrator must run them manually afterwards.

---

### Principle #6

Install UCS components through the App Center.

---

#### See also:

For more information about join scripts, see the following resources in *Univention Corporate Server - Manual for users and administrators* [2]:

- [How UCS systems join domains](#)<sup>38</sup>
- [Joining domains via Univention Management Console module](#)<sup>39</sup>
- [Join scripts / Unjoin scripts](#)<sup>40</sup>

## 3.3 Consequences of unfinished join scripts

Services won't work properly, or administrators can't manage them, if the join scripts didn't run during the package installation or upgrade.

---

### Principle #7

Verify status and version of the join scripts in the following situations:

- After installing software or apps.
  - After software or app updates.
  - When services aren't running as expected.
- 

To verify status and version of join scripts, run the command **univention-check-join-status**.

To run pending join scripts, use the command **univention-run-join-scripts** as described in [Subsequent running of join scripts](#)<sup>41</sup>. However, be careful with the `--force` option and the UCS server role on which you run the command.

---

### Principle #8

Never run **univention-run-joinscripts --force** on a *Primary Directory Node*.

The LDAP server doesn't work properly anymore and the repair is a lot of effort.

---

---

<sup>38</sup> <https://docs.software-univention.de/manual/5.0/en/domain-ldap/domain-join.html#linux-domain-join>

<sup>39</sup> <https://docs.software-univention.de/manual/5.0/en/domain-ldap/domain-join.html#linux-domain-join-umc>

<sup>40</sup> <https://docs.software-univention.de/manual/5.0/en/domain-ldap/domain-join.html#domain-ldap-joinscripts>

<sup>41</sup> <https://docs.software-univention.de/manual/5.0/en/domain-ldap/domain-join.html#domain-ldap-joinscripts-execlater>



## PRINCIPLES

The following list summarizes the principles described in the document for your reference. To jump to the location in the document for more context on the principle, click the title of the principle.

***Principle #1 (page 5)***

Don't manually edit configuration files that are under control of UCR or listener modules.

***Principle #2 (page 7)***

Use the standard software packages that UCS installs to provide core product capabilities.

Don't replace these software packages with software of your own personal taste.

***Principle #3 (page 8)***

Use the **univention-\*** tools to perform actions for installing, updating and removing software packages and apps on UCS.

***Principle #4 (page 9)***

Before installing software packages from third-party sources:

1. Always verify the App Center and the standard Univention software repositories, if the software is already available there.
2. Make sure that the packages don't overwrite existing packages.
3. Use **pip** only in virtual Python environments.

***Principle #5 (page 11)***

Don't run **univention-join** on a *Primary Directory Node*. It just skips.

***Principle #6 (page 12)***

Install UCS components through the App Center.

***Principle #7 (page 12)***

Always verify status and version of join scripts in the following situations:

- After installation of software or apps.
- After updates of software or apps.
- When services don't run as expected.

***Principle #8 (page 12)***

Never run **univention-run-joinscripts --force** on a *Primary Directory Node*.

The LDAP server doesn't work properly anymore and the repair is a lot of effort.



## BIBLIOGRAPHY

- [1] *Univention Corporate Server 5.0 Architecture*. Univention GmbH, 2023. URL: <https://docs.software-univention.de/architecture/5.0/en/>.
- [2] *Univention Corporate Server - Manual for users and administrators*. Univention GmbH, 2021. URL: <https://docs.software-univention.de/manual/5.0/en/>.



## INDEX

### B

Bugzilla

Bug #56790,4