What's new

UniconOS & Management 2611

Release date of UniconOS 2611: **2026-11-20**

> **Important:**

> **Breaking change for Azure Virtual Desktop (AVD) and Microsoft Entra ID:** The AVD client and the Entra ID sign-in on the device now use a new Microsoft Entra application, application ID `491bb6b0-7940-4e0f-bb52-b2989db8680b`. An AVD or Entra ID setup that works today can stop working after the update: sign-ins can fail, and Conditional Access policies, user assignments and allowlists that refer to the old application stop applying without any warning. Before you update any device, an administrator of your Microsoft Entra tenant must grant admin consent to the new application and add it wherever the old application is used. Then test the update with a pilot group. The old application, `65a544d1-2825-432e-9083-d73f9c786d92`, will be retired; devices that still use it then fail to sign in to AVD, even if nothing was changed on them. For the steps, see Moving the AVD client to the new Entra application.

> **Important:**

> UniconOS Management 2611 LTSR no longer includes Scout Console, Scout Report Generator, the legacy ELIAS and the Scout Command Interface (`scmd.exe`). Before you upgrade, read the Deprecation section of these release notes.

> **Important:**

> UniconOS Management 2611 needs Citrix® License Server 2607 (build 56200) or later. Upgrade your License Server before you upgrade UniconOS Management. With an older

License Server, Scout runs on its license grace period and stops serving devices when the grace period ends. For details, see Licensing.

New Features

Authentication & SSO

- **AVD client signs in with a new Microsoft Entra application**. The Azure Virtual Desktop client and the Entra ID sign-in on the device use a new Microsoft Entra application, so tenant administrators must prepare their tenant before the update, as described in the note at the top of this page. Users sign in once more after the update. The new device setting `OauthClientId` (section `AVD` of `/setup/terminal.ini`) can keep devices on the old application during a staged rollout or point them to your own app registration. For the steps, see Moving the AVD client to the new Entra application.
- **Imprivata XenDesktop® automation with desktop chooser**. After an Imprivata tap-in, UniconOS reads the user's Citrix desktops from the StoreFront™ stores in the Imprivata computer policy and starts the desktop, or shows a desktop chooser when the user has several desktops. Users with different desktops can now share the same boot-to-VDI device, and changes to their desktops or to the Imprivata policy apply at the next login without changes in UniconOS Management. XenDesktop automation now needs boot-to-VDI, and the desktops that are selected in a XenDesktop user policy no longer start. Before you update, see Imprivata XenDesktop automation. For the setup, see XenDesktop Automation and the Desktop Chooser.
- **Imprivata: log off instead of lock when users leave the device**. If the Imprivata policy sets **Inactivity lock or hot key behavior** to **Log off user and terminate SSO session**, a tap-out with the card, the hot key or inactivity now logs the user off UniconOS and disconnects the Citrix session, so the next person at a shared device can't reach it, while the user stays signed in to Windows and can reconnect later. UniconOS also applies the hot key, the inactivity time and the inactivity warning of the Imprivata policy with both options, and there's no setting for this in UniconOS Management. For details, see Lock or Log Off When Users Leave the Device.
- **Web applications open automatically after an Imprivata sign-in**. UniconOS now follows the Local Web Applications policies of Imprivata EAM 26.3 and later: right after an Imprivata sign-in, Chrome opens the web applications from the user's Imprivata policy, and the Imprivata browser extension signs the user in to them. Administrators choose the web applications per user in the Imprivata Admin Console, without changes in UniconOS Management. At a user switch, UniconOS closes the previous user's browser before it opens the web applications of the next user. For details, see Opening web applications automatically.

Input, accessibility & localization

- **On-screen keyboard that users can't close by accident**. A new setting removes the close button of the on-screen keyboard on the logon screen, the desktop and the lock screen, so that users of touch-only devices no longer lose their only keyboard with a stray tap, and the keyboard no longer has a key that quits it. On the device, the setting is **Hide keyboard close button** in the **Accessibility** section of the device configuration. In UniconOS Management Board, it is **Prevent closing the virtual keyboard** under **Keyboard and Mouse** > **Virtual keyboard**, for devices with UniconOS 2611 or later, and a user right of the same name under **Mouse/Keyboard** stops users from changing it on the device. For details, see Preventing users from closing the keyboard.
- **Hebrew, Korean and German (Neo 2) in the device configuration of UniconOS Management Board**. UniconOS Management Board now offers Hebrew and Korean in the desktop language list and German (Neo 2) in the keyboard layout list, in the device configuration and in configuration labels, so administrators can set them centrally instead of on each device. The REST API accepts these values too. UniconOS has supported Hebrew and Korean since 2607. For the settings, see Language and Device configuration / Keyboard and Mouse.

Connectivity & hardware

- **Several network connections at the same time**. Administrators can let UniconOS devices keep their Ethernet, Wi-Fi and mobile internet connections up at the same time, with the traffic on the preferred connection (Ethernet by default) and the other connections ready to take over, for example when a laptop is undocked. The settings **Allow parallel connections** and **Preferred connection type** are in UniconOS Management Board under **Device configuration** > **Network**, devices apply them after a restart, and the UniconOS desktop and Quick system access, also on the login screen and the lock screen, show each connection as connected. For details, see Parallel connections and Using Ethernet and Wi-Fi at the same time.
- **MAC address passthrough for Lenovo ThinkPad laptops on the HP USB-C Dock G5**. A docked Lenovo ThinkPad L15 Gen4 AMD or L16 Gen1 AMD now uses its own MAC address on the network instead of the dock's, also when it is docked while running. MAC-based network access control, DHCP reservations and the device's identity in UniconOS Management keep working when the laptop is docked; the feature needs MAC Address Pass Through in the BIOS and the new MAPT Implementations package in the image. For details, see Using MAC address passthrough.
- **eSIM profiles managed on the device**. Users add an eSIM profile in the device configuration under **Network** > **Mobile internet (WWAN)**, by scanning the operator's QR code with the

webcam or by typing the activation code, and delete it there again, while the device is also online over LAN or Wi-Fi. When a user connects, UniconOS switches to the right SIM slot and eSIM profile by itself, so devices with both a SIM card and an eSIM can use either one, one at a time. eSIM operations also work through a proxy, and diagnostic files now include the modem's firmware details. For details, see Mobile internet and eSIM.

- **Card reader of the Lenovo USB Wired Smartcard Keyboard 5000 detected**. UniconOS now detects the smart card reader built into the Lenovo USB Wired Smartcard Keyboard 5000 without a custom driver bundle. For details, see USB and smart card.
- **CUPS driver, PPD model and printer options in UniconOS Management Board**. In the **Printer** section of the device configuration and through the REST API, administrators can now set the CUPS driver of a printer, for example `zebra` for a Zebra label printer, and turn on the text filter separately. For devices with UniconOS 2601 or later, they can also set a PPD model and CUPS printer options such as `PageSize=A4`, so label sizes no longer need advanced file entries. For details, see Device configuration / Printer.

Deployment, VDI & migration

- **USB recovery image included in the AllPackages ZIP**. The AllPackages ZIP now also contains the USB recovery image (`.dd` file), and **Software** in UniconOS Management Board and ELIAS 18 import it into the container together with the software packages. Administrators can export an image for a USB recovery stick (StickWizz) from a new container right away, without downloading and importing the recovery image separately. For details, see Adding packages.

Device management & provisioning

- **Image management in UniconOS Management Board**. The new **Software** section of UniconOS Management Board takes over from ELIAS: administrators keep software packages in containers, build images with automatic dependency solving, sign them and manage certificates, with rights per container, and the activity log can record every change. The Software Management backend installs with Scout by default, and devices download the images from a web server or FTP server that publishes the container folders. For details, see Software.
- **Images held back when a package changes**. Software remembers the content of the package files that each image was saved with. If a package changes in the container folder afterwards, Software takes the image file out of the folder, so that devices keep their installed image until an administrator with the right to sign images accepts the change or the original file is back. Devices must download from a server that reads the container folders on every

request; mirrors, replicas and caches of image files aren't supported. For details, see Images held back after a package change.

- **Export a package from the Software library**. The menu of a package in the **Software library** now offers **Export**. It downloads the package version as one `.zip` archive with its features and signature files, which you can import into another container as it is. For details, see Exporting a package.
- **Firmware settings of an image**. The menu of a saved image offers **Firmware settings** with the protocol, server, path and image file to enter in the **Firmware** section of the device configuration, each with a copy button. Protocol and server appear once `DEVICE_DOWNLOAD_URL` in the configuration of the Software Management backend names the address of your web or FTP server. For details, see Configuring the devices.
- **Boot to VDI start scenario in UniconOS Management Board**. Administrators can now set the start scenario **Boot to VDI** in the **User authentication** section of the device configuration and in configuration labels, together with the Citrix store and the desktop or application that devices open after sign-in. Scout sends the setting to the devices, so Boot to VDI no longer needs advanced file entries, and existing configurations keep the normal desktop. For details, see Start scenario.
- **Configuration templates for start scenarios and kiosk mode**. Under **Device management** > **Configuration templates**, UniconOS Management Board offers ready-made templates for Boot to VDI, for the UniconOS desktop (Boot to apps) and for kiosk mode with Citrix Workspace™ app or the built-in browser, and administrators can add their own. A template is a configuration label that administrators copy, adapt, export or import and link to configuration rules, so a group of devices gets a complete start scenario or kiosk setup in one step. For details, see Configuration templates.
- **Kiosk Mode section removed from the device configuration**. The **Kiosk Mode** section of the UniconOS Management Board device configuration is no longer available, because UniconOS devices never applied its settings, and Scout no longer sends the file `/setup/kioskconfig.ini`. Kiosk mode set up with the advanced file entry `KioskMode` and the file `/setup/kioskmode.ini` works as before, and the kiosk configuration templates set it up this way, see Kiosk templates.
- **Browser prompt suppression in UniconOS Management Board**. Administrators can now turn on **Suppress browser prompts** and enter **Trusted browser origins** in the **User authentication** section of the device configuration and in configuration labels, instead of setting advanced file entries. On devices with UniconOS 2609 or later, Google Chrome and Microsoft Edge then no longer offer to save passwords, and Chrome, Edge and Firefox open Citrix Workspace app from the device's stores and the trusted origins without asking, so users

of kiosk and shared devices aren't interrupted during the Citrix sign-in. For details, see Browser prompts.

- **Scout Server checks the access password of protected OUs**. When a new device joins an OU that has an access password, or a hidden OU, or when a device moves itself into such an OU, Scout Server now checks the password itself. By default, also after the upgrade, it doesn't admit a device that sends no password: a new device starts in the default OU, and a moving device stays in its OU. The setup wizard of UniconOS 2611 and earlier, DHCP options, a GroupID in the image and Scout Cloud Gateway don't send the password, so place such devices with OU filter rules, or select **Accept and log** under **Preferences** > **Devices** to admit them as before, see Devices.

Platform & administration

- **UniconOS Management Board is the only management interface: Scout Console, Scout Report Generator, legacy ELIAS and scmd.exe removed**. The UniconOS Management 2611 LTSR installer no longer installs Scout Console, Scout Report Generator, the legacy ELIAS or the Scout Command Interface (`scmd.exe`), and an upgrade removes them after a notice that names their replacement. These end-of-life tools were installed for the last time by UniconOS Management 2609 (15.2609); you now manage devices and build images in UniconOS Management Board and automate tasks with the UniconOS Management Board REST API. For details, see Deprecation.
- **Citrix look and the product name UniconOS Management in UniconOS Management Board**. UniconOS Management Board now uses the Citrix colors, fonts and logos, also on its logon screen and in the **About** and **License information** dialogs, and the navigation menu and the browser tab show the product name UniconOS Management. Most texts that said eLux® or Scout now say UniconOS or UniconOS Management, for example **UniconOS Command Scheduler**, while service and component names such as Scout Server and Scout Board Database Layer stay the same. For details, see Main window.
- **Hide the side panel of UniconOS Management Board pages**. On the **Devices**, **Security & permissions**, **Preferences** and **Software** pages, administrators can hide the panel next to the navigation menu, so that lists and tables get the full width, and open it over the page when they need it. UniconOS Management Board remembers the choice for each section. For details, see Main window.
- **UniconOS Management Board REST API: device lists and device settings**. A new endpoint returns the devices of an OU, optionally with all sub-OUs, page by page and with only the fields a script asks for, for example for a nightly export of MAC addresses. A second endpoint reads and changes the device settings of **Preferences** > **Devices**, such as the default OU, the OU

filter mode and the name template. For details, see Loading the devices of an OU and Reading and changing device settings.

- **Restored or copied Scout databases work without an unlock step**. When Scout Server starts with a restored or copied Scout database, for example after a move to another SQL Server or in a test environment, it binds the license data to this database by itself and starts normally. Administrators no longer need a validation code to unlock the database, and Scout keeps checking the entitlement through the Citrix License Server. For details, see Restoring or copying the Scout database.
- **Stored passwords are encrypted with a data key of your installation**. Scout encrypts the passwords that it stores, for example in configuration labels, with a data key that each installation creates for itself and keeps in the Scout database, so a database backup contains this key. To keep the key apart from the database, you can store it in a key file on each Scout server instead. Scout Server converts passwords from earlier versions at every start. A later version will no longer read passwords in the format of earlier versions. For details, see Data key for stored passwords.
- **UniconOS Management Board and the REST API never return stored passwords**. Pages and REST API responses only show whether a password is set, and configuration exports don't contain stored passwords. An unchanged password is kept, a new one is stored encrypted, and a password can be cleared, so REST scripts that read a setting and send it back unchanged keep working. Label passwords sent through the REST API are now stored encrypted too, and a configuration label can no longer set the kiosk lock password. For details, see Stored passwords.
- **Scout configuration files protected from standard users**. Standard Windows users can no longer read or change the configuration files of Scout, for example `eluxd.ini` and `install.ini`, which can hold secrets. Administrators open these files in an editor that runs as administrator. For where the files are, see File path for Scout Server files.
- **Failed sign-ins limited**. After 10 failed sign-ins of one account from one address within 15 minutes, UniconOS Management Board refuses further sign-ins of this account from this address for up to 15 minutes, on its sign-in page and through the REST API, where the answer is HTTP status 429. Wrong device passwords are limited in the same way, so that passwords can't be found by trying many of them. For details, see Answers and errors.
- **UniconOS Management Board sessions end after inactivity**. UniconOS Management Board now ends a session after 30 minutes without activity and 12 hours after the sign-in at the latest. Administrators can change both limits, and the lifetime of REST API tokens, in the file `.env.custom` of UniconOS Management Board. After the upgrade, every administrator signs in again. For details, see Configuring session limits for UniconOS Management Board.

- **More label settings reach the devices**. Label settings that devices didn't receive before, for example Wi-Fi, mobile and VPN profiles, network drives, printers, SAML, user rights and the working days of power management, now reach them. After the upgrade, devices may apply label settings that they ignored before. For the settings, see Label settings on the devices.
- **File destinations and text files checked**. UniconOS Management Board now checks file destinations, INI file entries and the servers of RDP and AVD applications when you save them, also in configuration labels, imports and the REST API. It removes spaces at the start or end, adds the file name to a destination that ends with `/`, and refuses values that devices can't use, for example a path with a backslash, with the reason. For text files, it warns about Windows line breaks, a byte order mark and typographic quotes, and it can convert the line breaks. The built-in editor opens only UTF-8 text files. For what UniconOS Management Board refuses, see Upgrading to UniconOS Management 2611.
- **The Scout Board Database Layer accepts only allowed computers**. New installations and upgrades turn on the client check of the Database Layer, which then accepts connections on ports 22150 and 22151 only from the Scout server itself and the hosts in its list. An upgrade adds the hosts that it finds and keeps an existing client check setting, and a UniconOS Management Board on another computer must also be registered on the Scout server. For details, see Configuring the Database Layer client check.
- **Several Scout Servers on one Scout database set themselves up**. If several Scout Servers share one Scout database, installing, upgrading, repairing or adding a server no longer breaks the connections between them. The servers keep one shared key in the database, and every UniconOS Management Board finds the other servers by itself and receives their events; only the list of allowed hosts of the Scout Board Database Layer still needs the other servers' IPv4 addresses. The new **Node health** view under **Preferences** > **UniconOS Management instances** and the REST API health check show for each server whether its events reach UniconOS Management Board. For details, see Scout Server cluster.
- **Safer Active Directory sign-in to UniconOS Management Board**. UniconOS Management Board trusts the Windows certificate store for LDAPS, so domain controller certificates from an internal CA work without extra configuration. A domain controller that UniconOS Management Board finds through DNS is used only if its certificate shows that it is a domain controller of the domain. For details, see Domain controllers found through DNS.

Licensing

- **Citrix License Server 2607 or later required**. UniconOS Management 2611 needs Citrix License Server 2607 (build 56200) or later. Setup checks the version of the License Server on a new installation and on an upgrade and warns if it is older, and UniconOS Management

Board shows **License Server too old — upgrade to 2607 (build 56100) or later** instead of reporting the server as not reachable. Upgrade the License Server before you upgrade Scout: with an older one, Scout runs on its grace period and stops serving devices when it ends. For details, see Licensing.

- **UniconOS Management Board names a License Server that gives Scout no license**. If the License Server answers but holds no Scout licenses or isn't registered, UniconOS Management Board now shows **License Server gives UniconOS Management no license — check its UniconOS Management licenses and its registration** in the license information, on the dashboard and in System Status, instead of reporting the server as not reachable. The grace period stays the same. For details, see Licensing.
- **License Server certificate checked**. Scout remembers the certificate of the License Server when you configure it, and refuses a different certificate that Windows doesn't trust, so that nobody can pose as your License Server. After the certificate of the License Server changes, configure the License Server again in UniconOS Management Board. For details, see Configuring the License Server.
- **Citrix License Server is the license model for on-premises installations**. Scout Server, UniconOS Management Board and setup no longer support the retired license models Managed Service Provider, Enterprise Subscription and Software Maintenance. A new installation always uses a Citrix License Server, and an installation that still uses a retired model needs one after the upgrade, see Retired license models.
- **Activation keys and per-device licenses removed**. Scout no longer supports activation keys, per-device product licenses, temporary licenses, license threshold alerts or relocating devices together with their licenses. Devices with eLux RP 5 or earlier that request a license get no license from Scout. For details, see Licensing.
- **Setting "Automatically request licenses" removed**. The OU setting **Automatically request licenses** is no longer available in UniconOS Management Board. Every device gets the Fluendo Basic and Fluendo System licenses from Scout without this setting, as before with a Citrix License Server. For details, see Licensing.
- **External health check reports the license state**. When Scout has no valid license, the external health check of the UniconOS Management Board REST API now answers with the license state instead of HTTP 423 or 402. Monitoring that relies on these status codes must evaluate the license information in the answer instead. For details, see Monitoring with the health check.

Fixed issues

UniconOS Management 2611 (formerly Scout 15)

Fixed issues

Installation and services

- After a new installation of UniconOS Management 2609, Scout Keep Alive had no certificate on port 22124, so devices with keep-alive messages turned on couldn't report to Scout.
- In time zones east of UTC, the certificate that setup created for Scout Keep Alive became valid only hours after the installation, so the keep-alive port didn't work right away.
- A silent installation that needed a restart returned exit code 0 instead of 3010, so deployment tools didn't restart the computer.
- When Scout Server stopped because of a problem at start, the Windows event log showed the text of an unrelated system error, for example "Recursion too deep; the stack overflowed".
- The Windows Firewall rule **Scout Keep Alive** that setup created didn't open port 22124, so with the firewall on, devices couldn't send keep-alive messages to Scout. Setup now creates the rule for the port, and an upgrade corrects an existing rule.
- An upgrade reset `Encrypt`, `TrustServerCertificate` and `MultiSubnetFailover` that you had set by hand, for example `MultiSubnetFailover=1` for an SQL Server Always On listener. An upgrade now takes these values from `eluxd.ini`, and a silent upgrade takes the values of Scout Keep Alive from `scoutkeepalive.ini`. If you set them by hand, set them in `eluxd.ini` too.
- The Scout Board Database Layer and Scout Keep Alive services could stop unexpectedly when they wrote a message to the Windows event log, for example when they started.
- The setup wizard closed without a message when you went back from the page for the device password and then clicked **Next** on the database page again.
- If setup couldn't read the version of an existing Scout database, for example because the account had too few rights, it could delete the tables of the database and create them again, so that devices, OUs, configurations and administrators were lost. Setup now stops with an error and never deletes tables.
- If setup couldn't connect to the SQL Server or couldn't read the Scout database, it continued and failed later without the reason, and a failed new installation could leave a half-created database that blocked the next attempts. Setup now stops at the database step and shows the server, the database, the account and the SQL error.

- With Windows authentication and a service account that isn't an administrator of the Scout server, setup couldn't create the tables of a new Scout database and failed with "Invalid object name 'Administrator'".
- Every update and repair of UniconOS Management Board replaced the list of sign-in domains in `DOMAINS` with the domain of the server, so users of the other domains could no longer sign in. Setup now keeps the list.
- If the Scout database wasn't available yet when Scout Server started, for example because SQL Server on the same computer started later after a restart, Scout Server stopped during its start, sometimes with a crash dump, and stayed stopped. Scout Server now waits longer for the database. If it still can't connect, it logs the database server and name and stops with exit code 1017.
- When the certificate authority of Scout was renewed, only one of several Scout Servers on one Scout database gave itself a new certificate, so devices that check the server certificate couldn't connect to the other servers until these restarted.
- During an upgrade that converted the Scout database, the Scout Board Database Layer could stop working while Windows still showed it as running, and its service couldn't be stopped.
- After the Scout Server service had stopped, its process could stay for up to a minute, so a restart or an update right after the stop could fail.
- After a database error, Scout Keep Alive could stop, or it stopped updating its heartbeat so that Scout showed it as stopped, and it answered the keep-alive messages of devices with an error.
- On UEFI devices, PXE recovery from Scout started the recovery system of eLux® RP 6 instead of the recovery system of UniconOS.

Licensing

- On installations that use a Citrix® License Server, devices refused firmware that was built more than 12 months after the installation had started to use the License Server, because Scout sent them a subscription end date that never moved. The update log of the device showed "cannot update … (need subscription MM/YYYY)", and the update ended with "no credits (250)". Scout now sends a new end date with every license request.
- After the license grace period had ended, UniconOS Management Board kept reloading its license page, so you couldn't configure the License Server there.
- Scout Server could stop unexpectedly when it couldn't reach the License Server to send its usage data.
- After you had configured the License Server, UniconOS Management Board could show the old license state for up to 30 minutes.

Sign-in to UniconOS Management Board

- While a maintenance window was active, administrators who got their rights only through an Active Directory group could still sign in to UniconOS Management Board, where every dashboard widget showed "Error!".
- When two SAML sign-ins to UniconOS Management Board started at the same moment, for example in two browser tabs, one of them could fail.
- With OpenID Connect sign-in, a short outage of the identity provider sent every page load of UniconOS Management Board to a sign-in page that couldn't complete.

Configuration and rights

- Devices set to Russian didn't get a working Russian locale, because Scout sent the desktop language in a format that UniconOS doesn't accept, and the REST API refused Russian. Scout now sends `ru_RU`, and the upgrade corrects existing configurations and labels.
- In **Power management** > **Edit profiles**, **Screen saver after** was shown as turned off for a new OU, but after you saved a profile change, it was turned on and applied to the devices.
- Administrators who may edit administrators but not their menu rights couldn't save any change to another administrator.
- When you changed the device password in UniconOS Management Board, Scout Server used the new password only after it restarted, and devices with their own OU or device configuration didn't get it with their next configuration.
- Devices that confirmed a new configuration too late got the complete configuration again at every start, which overwrote local settings such as the monitor layout, the keyboard or the volume, also with **Retain local device configuration (unlocked fields)** turned on. Such devices recover at their first contact after the upgrade.
- A **Files** list of the advanced device configuration couldn't be saved any more once its files, including the inherited ones, had more than about 768 KB; every save failed with "Field value too long".
- The `<COUNT>` variable of the device name template started again at its first value after Scout Server restarted, so new devices got numbers that other devices already had.
- Scout sent the bitrate of serial printers in a key that UniconOS doesn't read, so serial printers ran at the speed of the serial port instead of the bitrate set in UniconOS Management Board.
- For a new device, configuration rules on the supplier, the serial number or the device type that Scout stores for the device matched only from its second configuration, unless the device name template used one of these properties.

- A printer driver name that was set outside UniconOS Management Board, for example with the legacy Scout Console, was shown as **PCL2** and replaced when you saved the printer.
- In **Preferences** > **Commands Templates**, switching off the last option of a template that was still on wasn't saved.
- For a configuration without display settings, for example after a damaged import, the display settings couldn't be saved, and turning off the inheritance for an OU or a device below it failed with "cannot copy Screen configuration".
- Changes that you saved under **Preferences**, for example for new devices, the OU filter, the device names or the logging, took effect in Scout Server only after it restarted.
- Turning off the Scout Server log under **Preferences** > **Logging** turned off only the error messages, and Scout Server kept writing its other messages.
- Two administrators who edited the same configuration at the same time, in UniconOS Management Board or through the REST API, could overwrite each other's changes, and UniconOS Management Board showed "Internal server error" instead of saying that someone else was editing.
- Administrators without the right to edit the base configuration saw "Ooops. Something went wrong" when they opened **Device management** under **Devices**.
- Editing a file of a **Files** entry that wasn't UTF-8 text, for example a certificate, in the built-in editor damaged the file.
- The VMware Horizon option to hide the connection bar automatically didn't reach the devices.
- The Evidian shared secret of an OU or device configuration was saved in the wrong place, so it didn't take effect.
- With SAML sign-in, Scout sent the smart card certificates of the configuration to the root folder of the devices.
- If many devices requested their configuration at the same time, for example after Scout Server had restarted, a device could get the scheduler jobs or the keyboard layout of another OU.

Configuration labels and imports

- Importing a configuration showed success even when some of its sections weren't applied.
- If Scout Server couldn't read one value of a configuration label, for example a keyboard setting of the wrong type or a file entry that isn't Base64, devices didn't get the rest of the label.
- Importing an OU configuration didn't restore the monitor layout of the OU or whether the OU inherits it.

- The configuration label editor lost some of the changes that you applied together, for example two desktop pictures or a changed monitor layout.
- Logon information texts of a configuration label lost everything written in angle brackets, for example `<Company>`.
- An environment variable with **Encrypt value** in a configuration label reached the devices as a changed value at almost every configuration build, although nobody had changed it, so the devices applied it again each time.

Devices, diagnostics and commands

- When several new devices contacted Scout Server for the first time within the same second, Scout Server sometimes didn't save some of them. These devices didn't appear in UniconOS Management Board, and commands didn't reach them.
- If a device needed long to deliver its diagnostic files, UniconOS Management Board reported a failure and never delivered the files, and requesting them again started another collection whose files were never shown.
- Sorting the device list by the **OU Path** column made the list fail to load.
- Devices that waited for a custom command, or that the command job had started with Wake on LAN, showed **Command running** before they got the command.
- On the **Command History** page, **Refresh view**, **Terminate** and **Purge** didn't reload the list, so new commands and their progress appeared only after you opened the page again.
- In the device list, the filters for notifications such as "Software update on next start" and for the relocation server found no devices with "No" outside the first filter group, and "is not" worked like "is".
- Grid layouts, filters and sort orders carried over to the next administrator who signed in to UniconOS Management Board in the same browser.
- Moving an application out of **Lost & Found** failed with "Source application not found".

REST API

- Sending an empty list of USB rules to clear all rules returned HTTP status 500, and the rules stayed.
- The roaming switch of mobile (APN) profiles and two switches of LAN profiles weren't saved as sent, and LAN and Wi-Fi profiles that were read and sent back unchanged were refused.
- A printer read through the REST API couldn't be sent back unchanged, because the REST API refused the empty `DriverName` and `TpClass` values that it returned.

UniconOS 2611

Release date **2026-11-20**

Fixed issues

Authentication, Imprivata & smart card

- On boot-to-VDI devices with Imprivata, a card tap right after a virtual desktop session ended could be ignored, and the login screen showed "The operation was canceled because it was no longer needed."
- With Imprivata, the virtual desktops that are assigned to the user in Imprivata appeared on the desktop only after the user started another application, and a newly assigned desktop didn't start automatically at the first login.
- Imprivata card login didn't work with the HID OMNIKEY 5022 CL card reader.
- The domain list of the Imprivata sign-in screen showed an AD directory twice, with its DNS name and its NetBIOS name.

Display, monitors & graphics

- After a laptop was undocked, docked or woken up while the screen was locked, the unlock dialog could stop accepting keyboard input, so that only a restart helped.
- After the device woke up from sleep mode, a Chrome window that the user moved could jump back to its previous position for about 20 seconds.
- With some monitors, for example certain LG Ultra HD models, users couldn't change the monitor layout or open the device configuration from Quick system access.
- On the Lenovo ThinkPad E14 Gen 7 (type 21SX), mouse clicks could stop reaching the application windows after the lid was closed and opened.

Audio, peripherals & collaboration apps

- The mute button or the microphone boom arm of a USB headset also muted or unmuted the microphone in UniconOS, so that users could stay muted after a call although the headset and the softphone showed unmuted.
- On devices with an Intel SOF sound card, for example the Lenovo ThinkPad E14 Gen 7, there was no audio output after a 3.5 mm headset was unplugged while a DP or HDMI monitor was connected.

Hardware support

- On the Lenovo ThinkCentre M90n-1, the onboard network port (Realtek RTL8111/RTL8168) didn't work.
- On the Fujitsu ESPRIMO P5010n, the installed UniconOS didn't start, and the device reported "No Bootable Device".
- On the Lenovo ThinkPad L14 Gen 2, the mobile broadband modem always used the eSIM slot, so mobile internet didn't work with a SIM card.
- Wi-Fi didn't work on devices with an Intel Wi-Fi 6 AX203 adapter, because firmware files were missing.
- Wi-Fi didn't work on the HP t730 with a Broadcom BCM43228 adapter, because a firmware file was missing.

Device management

- If the name of the Scout Server couldn't be resolved when the device started, for example because the network wasn't ready yet, the device didn't contact Scout again until it was restarted.
- If Scout Server couldn't be reached when a device without a configuration started, for example after an update with **Format system partition before update**, the device stayed without a configuration and showed an empty desktop.
- After a restart while Scout couldn't be reached, the device showed its license as expired.
- When the built-in browser ran in its sandbox, its log files and crash dumps were lost, so they were missing from the diagnostic files.

Boot, shutdown & session stability

- Application starts could time out, and a restart or shutdown could take up to 30 minutes.
- After a laptop woke up from sleep mode, closing the lid could no longer put it to sleep.
- After a login or a restart, the system bar was sometimes missing.
- A notification, for example from Microsoft Teams in Chrome, could no longer be closed after it changed its size.

Migration (igel2UniconOS)

- igel2UniconOS migration failed on IGEL OS 11.08 devices while it prepared the EFI partition.

Applications & browser sessions

- With **Show last user** on for a StoreFront™ application, the logon dialog didn't remember the user name.
- With Persistent Home, Microsoft Edge didn't keep the user's data after a log off.

Security hardening

- Google Chrome was updated to 157.0.8103.78.
- Microsoft Edge was updated to 156.0.4314.58.
- OpenVPN was updated to 2.5.11-0ubuntu0.22.04.6 (fixes CVE-2026-84471, CVE-2026-84732 and CVE-2026-84790).
- BlueZ was updated to 5.64-0ubuntu1.6 (fixes CVE-2026-19774, CVE-2026-75032, CVE-2026-80185, CVE-2026-80186 and CVE-2026-85218).
- Security updates for brotli due to the critical vulnerability CVE-2026-111497.
- The AllPackages container archive contains Zoom Client for Linux 7.2.5.6661, which replaces version 7.0.0.1666 and its outdated embedded browser components.
- The Linux kernel was updated to 6.18.59 (fixes CVE-2026-98432).
- Firefox ESR was updated to 153.6.0.
- libsoup3 was updated to 3.4.4-5ubuntu0.9 (fixes CVE-2026-1801, CVE-2026-4271, CVE-2026-5119, CVE-2026-6324, CVE-2026-12549, CVE-2026-66339, CVE-2026-77014, CVE-2026-77680, CVE-2026-85197 and CVE-2026-85534).
- squid was updated to 5.9-0ubuntu0.22.04.8 (fixes CVE-2026-61642).
- Security updates for aom due to the critical vulnerabilities CVE-2026-56208–56211.
- Security updates for atril due to the critical vulnerability CVE-2026-46529.
- Security updates for attr due to the critical vulnerability CVE-2026-54371.
- Security updates for bzip2 due to the critical vulnerability CVE-2026-42250.
- Security updates for cifs-utils due to the critical vulnerability CVE-2026-104715.
- Security updates for coreutils due to the critical vulnerability CVE-2025-5278.
- Security updates for curl due to the critical vulnerabilities CVE-2026-8932, CVE-2026-11856, CVE-2026-18924, CVE-2026-80230, CVE-2026-82209, CVE-2026-86409, CVE-2026-86414, CVE-2026-87862, CVE-2026-87864, CVE-2026-92388–92389, CVE-2026-92391, CVE-2026-94266.

- Security updates for expat due to the critical vulnerabilities CVE-2026-32776–32778, CVE-2026-50219, CVE-2026-56131–56132, CVE-2026-56403–56412, CVE-2026-66046, CVE-2026-72522, CVE-2026-76641, CVE-2026-76957.
- Security updates for ffmpeg due to the critical vulnerabilities CVE-2026-8461, CVE-2026-12706, CVE-2026-30754, CVE-2026-30999, CVE-2026-38344, CVE-2026-38346–38350, CVE-2026-58049, CVE-2026-64830, CVE-2026-64832–64835, CVE-2026-65703–65706, CVE-2026-66036, CVE-2026-66038–66039, CVE-2026-70628–70632, CVE-2026-75141–75144, CVE-2026-75146–75147.
- Security updates for freetype due to the critical vulnerability CVE-2026-95512.
- Security updates for ghostscript due to the critical vulnerabilities CVE-2026-39919, CVE-2026-101258, CVE-2026-103226.
- Security updates for glib2.0 due to the critical vulnerabilities CVE-2026-15588, CVE-2026-16118, CVE-2026-58010–58016.
- Security updates for glibc due to the critical vulnerabilities CVE-2026-6368, CVE-2026-6791, CVE-2026-19542, CVE-2026-77117, CVE-2026-80489.
- Security updates for gst-plugins-bad1.0 due to the critical vulnerability CVE-2026-19387.
- Security updates for gst-plugins-base1.0 due to the critical vulnerability CVE-2026-18297.
- Security updates for gst-plugins-good1.0 due to the critical vulnerabilities CVE-2026-17072, CVE-2026-18295–18296, CVE-2026-18298–18299, CVE-2026-18649, CVE-2026-73433–73434, CVE-2026-88914.
- Security updates for gvfs due to the critical vulnerabilities CVE-2026-84268, CVE-2026-88924.
- Security updates for libarchive due to the critical vulnerability CVE-2026-16517.
- Security updates for libevent due to the critical vulnerabilities CVE-2026-63381–63385, CVE-2026-63388.
- Security updates for libgcrypt20 due to the critical vulnerability CVE-2024-2236.
- Security updates for libpcap due to the critical vulnerability CVE-2026-0799.
- Security updates for libpng1.6 due to the critical vulnerabilities CVE-2026-40930, CVE-2026-46675.
- Security updates for librabbitmq due to the critical vulnerabilities CVE-2026-59986, CVE-2026-61547.
- Security updates for libsoup2.4 due to the critical vulnerabilities CVE-2026-1467, CVE-2026-1539, CVE-2026-1801, CVE-2026-12549, CVE-2026-66339, CVE-2026-77014, CVE-2026-77680.

- Security updates for libssh due to the critical vulnerabilities CVE-2026-59843, CVE-2026-59845–59848, CVE-2026-59850.
- Security updates for libx11 due to the critical vulnerabilities CVE-2026-88806, CVE-2026-94283–94285.
- Security updates for libxml2 due to the critical vulnerabilities CVE-2026-74860, CVE-2026-76781, CVE-2026-86138, CVE-2026-86140, CVE-2026-86142–86144.
- Security updates for libxpm due to the critical vulnerability CVE-2026-94287.
- Security updates for libxtst due to the critical vulnerability CVE-2026-94286.
- Security updates for ncurses due to the critical vulnerability CVE-2025-6141.
- Security updates for openssh due to the critical vulnerabilities CVE-2026-73281–73283, CVE-2026-106552–106553, CVE-2026-106555, CVE-2026-106582–106587.
- Security updates for openssl due to the critical vulnerabilities CVE-2026-35189, CVE-2026-54872, CVE-2026-54874, CVE-2026-63072, CVE-2026-63074, CVE-2026-63076, CVE-2026-75803, CVE-2026-75805–75806, CVE-2026-77696, CVE-2026-84782.
- Security updates for p11-kit due to the critical vulnerabilities CVE-2026-13757, CVE-2026-18938.
- Security updates for perl due to the critical vulnerabilities CVE-2026-12087, CVE-2026-13221, CVE-2026-15534, CVE-2026-19487, CVE-2026-57432–57433.
- Security updates for policykit-1 due to the critical vulnerability CVE-2026-85498.
- Security updates for poppler due to the critical vulnerabilities CVE-2026-93312–93314, CVE-2026-102620–102621, CVE-2026-107705.
- Security updates for python3.10 due to the critical vulnerabilities CVE-2026-4360, CVE-2026-15308.
- Security updates for sqlite3 due to the critical vulnerability CVE-2026-39113.
- Security updates for srt due to the critical vulnerabilities CVE-2026-55868–55869.
- Security updates for systemd due to the critical vulnerabilities CVE-2026-15059, CVE-2026-16742.
- Security updates for vim due to the critical vulnerabilities CVE-2026-73071–73074, CVE-2026-73076–73078.

Known issues

UniconOS Management 2611 (formerly Scout 15)

Limitations and known issues

- OpenID Connect as an authentication method in the device configuration dialog will only be supported from UniconOS version UniconOS 2601 onwards.
- SAML as authentication method for eLux® require additional setup via Advanced File Entries; simplified configuration will follow in a future release:

  OKTA /setup/terminal.ini | Security | AuthType | 15

  /setup/terminal.ini | SAML | SsoUrl | <URL> /setup/terminal.ini | SAML | Issuer | <URL>

  /setup/terminal.ini | SAML | Audience | <URL> /setup/terminal.ini | SAML | CertificatePath | <Path>

  Please transfer the SAML certificate to the configured certificate path via file transfer.
- Entra ID as authentication method in the device configuration is currently only supported for AVD use cases.
- Some texts on the **Software** pages of UniconOS Management Board still use the former product names eLux and Scout.
- Several Scout Servers on one Scout database: a new installation allows only the server itself in the list of allowed hosts of the Scout Board Database Layer, and an upgrade adds only the servers that it finds connected. Each server refuses the UniconOS Management Boards of servers that aren't in its list. On every server, add the IPv4 addresses of all other servers to `DBLayerAllowedClients`, see Scout Server cluster.
- Several Scout Servers on one Scout database: if more than one server runs Software Management, each server lists the containers of the other servers but can't use them. Install Software Management on one server only.
- Several Scout Servers on one Scout database: if two servers name new devices at the same moment, both devices can get the same number from `<COUNT>` in the name template, see Device names.
- In the **User authentication** section of the device configuration, the information below the browser prompt settings says that Firefox suppresses prompts on every site. Like Google Chrome and Microsoft Edge, Firefox opens Citrix Workspace™ app without asking only on trusted origins.

- The **Appearance** entry in the user menu of UniconOS Management Board offers the color schemes **Dark**, **High contrast** and **Use system setting**, which follows the light or dark mode of the operating system. These color schemes aren't released yet, and not every page follows them. Use **Light**, the default.
- Keyboard users can't move into the menus that open when you right-click an OU or a Dynamic Device Group, and Esc doesn't close these menus.
- After a silent upgrade with a new service account password, setup keeps the old password in `install.ini`. A later silent upgrade without credentials gives the Scout services the old password, and they don't start. After a password change, pass the service account credentials with every silent upgrade.
- A silent repair can change a database password that contains square or curly brackets, for example `}`, in `eluxd.ini` and `dblayer.ini`, and the Scout services then can't sign in to the database.
- After an update, the sign-in page of UniconOS Management Board can say that this UniconOS Management Board is not registered with the Scout server, and **Software** can show no containers. The update gave UniconOS Management Board and the Software Management backend new keys but didn't register them on the Scout server. Register the keys that the log files name, see Allowing a UniconOS Management Board on another computer and Software Management backend on another computer.
- If Scout Server refuses the certificate of the License Server, **System Status** still shows the License Server as reachable. The **License information** dialog shows the refusal.
- With several Scout Servers on one Scout database, configuring the License Server again in UniconOS Management Board saves the new certificate of the License Server on one Scout Server only. The other Scout Servers keep refusing a new certificate that Windows doesn't trust. On each of them, enter the thumbprint of the new certificate as `LSCertHash` in the `[LS]` section of `eluxd.ini`, and restart the Scout Server service, see Licensing.
- An upload in **Software** must be finished within one hour after it starts. On a slow connection, the upload of a large file, for example the AllPackages archive of a UniconOS version, can fail. Upload large files from a computer with a fast connection to UniconOS Management Board.
- If you rename a container in **Software** while an archive is imported into it, and another container gets the old name right away, the rest of the import can go into the other container. Don't rename a container while an import into it is running.
- In the **Display** section of the device configuration, **Add monitor** adds no monitor to a configuration that has no monitor layout, for example one damaged by an earlier configuration import. OUs and devices below such a configuration get a layout with one monitor when you

turn off **Use parent device configuration** for them. To repair the configuration itself, contact Citrix® Support.

- If you turn off **Use parent device configuration** for an OU or a device below a configuration that has no monitor layout, and add or change monitors in the same save, a monitor can be stored twice. Save the change of **Use parent device configuration** first, and then change the monitors.
- When you move an application to another OU in UniconOS Management Board, the devices below the OU that the application left keep the application until their configuration is built again for another reason. If you move an application of the base configuration, this applies to all devices.
- If several configuration labels match one device, the device gets the Active Directory domains, user variables, USB rules and host names of all these labels, and an entry that two of the labels contain reaches the device twice. If two of the labels contain COM port settings, the settings of the second label go to the next COM port. Set each of these lists in only one of the labels that match a device.
- Administrators who sign in through the same administrator group share one set of saved grid layouts, filters and sort orders.
- A printer change through the REST API that doesn't contain `TextFilter` turns off the text filter of the printer, so send `TextFilter` with every printer change. Send printers one by one: the complete `printerConfig` object, sent back as the REST API returned it, is refused with HTTP status 400.

UniconOS 2611

Release date: **2026-11-20**

Limitations and known issues

- OIDC and SAML authentication methods for UniconOS require additional setup via Advanced File Entries. For further details, please refer to the UniconOS Management release notes.
- With Imprivata XenDesktop® automation, the first card tap after a device restart sometimes finds no desktops and shows an error panel; **Retry** reads the desktops again.
- With Imprivata XenDesktop automation, the device notices the end of a desktop session only when Citrix Workspace app has closed, which can take about 20 seconds after a Windows sign-out; a card tap before that is handled as a tap on the running session.
- With Imprivata XenDesktop automation, starting a desktop, or reconnecting it after the device is unlocked, can sometimes take much longer than usual.

- With Imprivata XenDesktop automation, after a user enrolls a new card at the login screen, the desktop chooser can show no desktops, and an error panel appears after about 90 seconds. Tap the card again to show the desktops.
- With Imprivata web applications that open automatically after sign-in, UniconOS closes the browser at every user switch, even if the Imprivata policy is set to keep applications open on a user switch.
- If a Chrome application from UniconOS Management starts automatically in the same session as the Imprivata web applications, or the user opens Chrome while they are prepared, all pages open in the same Chrome browser, and the command line parameters of the Imprivata computer policy, such as `--start-fullscreen`, may not apply. In rare cases, a Chrome page that opens while the web applications are prepared is closed again, and the user has to open it again.
- With browser prompt suppression on, Microsoft Edge can still offer to save passwords in a profile that is signed in with a Microsoft account.
- With **Hide keyboard close button** turned on, the close button of the on-screen keyboard reappears when a user taps the predefined snippet **Example** on the keyboard's snippets page, and the user can then close the keyboard. Tapping **Example** again hides the close button.
- Tapping the predefined snippet **Example** on the snippets page of the on-screen keyboard makes the keyboard flicker and close. The **Example** snippet isn't supported.
- After a user resized or moved the on-screen keyboard, turning off **Show on screen keyboard** doesn't close it.
- After a change to a lower display resolution, the on-screen keyboard can disappear, depending on its position on the screen.
- On boot-to-VDI devices with Citrix Workspace app 2608, a log off at a tap-out can leave an empty Citrix Workspace app logon dialog on the Imprivata login screen for some time. As a workaround, set the advanced file entry `StorebrowseIPCMode=selfservice` in section `ICADefaults` of `/setup/sessions.ini`.
- With the start scenario Boot to VDI and Citrix Workspace app 2608, the next sign-in after a virtual desktop session ends doesn't start a new session and ends with "Your virtual desktop timed out". A Citrix Workspace app logon window can also appear over the login screen. Restart the device to sign in again. As a workaround, set the advanced file entry `StorebrowseIPCMode=selfservice` in section `ICADefaults` of `/setup/sessions.ini`, and restart the devices; sign-ins then take longer.

- After a device moves to the new Microsoft Entra application of the AVD client, a session that was locked with an Entra ID sign-in before the move can't be unlocked. Users log off from the lock screen and sign in again.
- After the device wakes up from sleep mode, **Mobile internet (WWAN)** can be missing in **Network** and in **Quick system access** until the device restarts.
- In the **Modem & eUICC information** dialog, **Delete** is available even if the user rights don't allow users to delete network profiles, so users can delete an eSIM profile from the eSIM chip.
- A CIFS drive with a user name and a password from Scout doesn't mount if the password contains a non-ASCII character, such as ä, ö, ü or ß, or is longer than 37 characters, although **Test** succeeds. Use a password of up to 37 ASCII characters, or Kerberos.
- If a CIFS drive can't be mounted, for example because of a wrong password, each attempt to open it sends about 80 failed sign-ins to the server, which can lock the user account. The user sees the message "Too many levels of symbolic links".
- With user authentication, the files that users save in their home folder, for example downloads, are kept in the device's memory. Large downloads can fill this memory, and local applications then don't start until the device restarts.
- On devices with 4 GB of RAM, Zoom Client for Linux can close unexpectedly during a meeting, because the device runs out of memory.
- The setup wizard checks the access password of a protected OU on the device, but doesn't send it to Scout. With UniconOS Management 2611, which checks this password by default, a new device that joins a protected or hidden OU through the setup wizard starts in the default OU. Place such devices with OU filter rules or move them after they have joined, or select **Accept and log** under **Preferences** > **Devices** in UniconOS Management Board, see Devices.
- If the **Name** of an Active Directory domain in the user authentication settings contains a dot, for example `corp.local`, the device drops the settings of this domain, and users can't choose it on the sign-in screen. Use only letters and digits in the name, for example `corp`. The entries in **Server, Server list or Domain** aren't affected.
- If a printer with the connection type **Serial (COM2)**, **Serial (COM3)** or **Serial (COM4)** has the bitrate **<System>**, the device can stop setting up its printers while it starts and then appears offline in UniconOS Management Board until it is switched off and on again. In UniconOS Management Board, select a **Bitrate** other than **<System>** for such a printer. With UniconOS Management 2609 or earlier, this applies to every printer on COM2, COM3 or COM4, because these versions don't send the bitrate in a form that UniconOS reads.
- If devices check the signatures of image definition files, devices in a time zone ahead of UTC can refuse an image that was signed within a few hours after its signing certificate became

valid. The update log then shows "no valid CA cert found". Use a signing certificate that has been valid for at least one day before you sign images.

Hardware limitations

- Monitor clone mode is not supported for more than two monitors.
- Disk encryption is only supported for devices with TPM 2.0 module.
- Sleep mode (Suspend to RAM) is not yet available for devices with eKabini chipsets.
- Multi-monitor setup in clone mode requires uniform resolution of all connected monitors.
- Wake on LAN via USB-C docking station is not functional.
- Dell Pro Micro TC Q9M1260: Pivot is not functional.
- Dell Pro 14 Plus PB14250 Intel: Microphone input via 3.5 mm audio jack is not functional.
- Dell Pro Thunderbolt 4 Smart Dock – SD25TB4: Firmware update is not functional.
- Dell Latitude 3410, 3420: Bluetooth is not functional.
- Dell Latitude 5540: eSIM is not functional. Physical SIM cards are not affected.
- Fujitsu FUTRO devices: UEFI updates are only supported for devices without disk encryption via TPM2.
- Fujitsu devices: BIOSset tool for BIOS configuration is not functional.
- HP mt645 G8: LTE is not functional.
- HP mt645 G7: Sleep mode (Suspend to RAM) is not functional.
- HP mt46: LTE is not functional.
- Lenovo ThinkPad T14 Gen7 - AMD Ryzen 5 Pro 215: LTE is not functional.
- Lenovo ThinkPads Gen4: LTE is not functional.
- Lenovo ThinkPads Gen4 AMD: Bluetooth is not functional.
- Lenovo ThinkCentre M90n-1: PXE boot is not functional. Workaround: add `pcie_port_pm=off` to the kernel command line of the PXE boot configuration.
- LG gram Mobile Thin Client 17ZT90P: LTE is not supported.
- LG Ultrabook 14UT70Q: Sleep mode and sound via HDMI are not functional.
- MOStron Touch –Panel PC TSPC5-24: BIOS update via UniconOS Management is not functional.
- NEXterminal XI DP/DP2: Internal speaker is not functional.

Misc - before you start

License Server 2607 or later

UniconOS Management 2611 needs Citrix License Server 2607 (build 56200) or later. If you upgrade from an earlier version and your License Server is older, upgrade the License Server first. Setup warns you if the License Server is older or if it can't read its version. With an older License Server, Scout runs on its license grace period after the upgrade and stops serving devices when the grace period ends. For details, see Licensing.

Retired license models

From UniconOS Management 2611, Scout no longer supports activation keys and the license models Enterprise Subscription, Managed Service Provider and Software Maintenance. A new installation always uses a Citrix License Server. If your Scout still uses activation keys or one of these license models, configure a Citrix License Server after the upgrade: until then, UniconOS Management Board opens only its license page, and devices can't connect to Scout Server. For details, see Licensing.

Upgrading to UniconOS Management 2611

Check these changes before you upgrade. They can change what UniconOS Management Board accepts and what devices receive:

- If Scout uses a Citrix License Server, upgrade Scout to 2611 before you update the devices to UniconOS 2611. UniconOS Management 2508 LTSR to 2609.1 can give the devices an end date of their software maintenance that no longer moves, so devices refuse firmware that was built more than 12 months after Scout started to use the License Server. The update log of such a device shows `need subscription`, and the update ends with `no credits (250)`. After the upgrade, devices get a new end date with their next license request, and then you can run the update again, see Upgrading Scout before the devices.
- Scout Server now checks the access password of a protected or hidden OU when a new device joins the OU or a device moves itself into it. By default, also after the upgrade, it doesn't admit a device that sends no password: a new device starts in the default OU, and a moving device stays in its OU. The setup wizard of UniconOS 2611 and earlier, DHCP options, a GroupID in the image and Scout Cloud Gateway don't send the password. If new devices join protected or hidden OUs this way, place them with OU filter rules, or select **Accept and log** under **Preferences** > **Devices**, see Devices.

- Devices must download their images from a server that reads the container folders of **Software** on every request. Mirrors and replicas of the container folders, and caches that keep image files (`.idf`), aren't supported, see Publishing a container to devices.
- By default, devices don't check signatures before a firmware update, so they install images and packages even if they are unsigned or were changed. To make devices refuse them, sign your images and select **Image definition files (.idf)** and **UniconOS software packages** under **Security** in the **Firmware** section of the device configuration, see Checking signatures.
- Scout encrypts stored passwords with a data key of your installation and converts passwords from earlier versions at every start of Scout Server. The data key is stored in the Scout database, so a database backup contains it. Protect database backups like the database itself. For the optional key file, see Data key for stored passwords.
- A label password that Scout Server can't read is left unchanged and listed in its log. Scout leaves out the label setting that holds it, so the device configuration applies for this setting, and the rest of the label still reaches the devices. A later version will no longer read passwords in the format of earlier versions.
- Configuration label settings that devices didn't receive before now reach them, for example Wi-Fi, mobile and VPN profiles, network drives, printers, SAML, user rights and the working days of power management. A label no longer removes network profiles, network drives, printers or background pictures of the device configuration, see Label settings on the devices. Check your labels before the upgrade. A label whose power management has no working day selected keeps the working days and hours of the OU. To make sure that devices get the settings of a label, open the label after the upgrade, choose **Edit**, and then choose **Apply**. You don't need to change anything.
- The upgrade doesn't correct these settings that earlier versions of UniconOS Management Board saved wrongly. Correct them after the upgrade:
  - The Evidian shared secret of an OU or a device that you saved in UniconOS Management Board didn't take effect, because it was saved in the wrong place. Enter it again once, and save. Devices get it with their next configuration.
  - A configuration label with **Diagnostics** settings sets both **Enhanced log level** and **Enhanced logging for smart card support**, even if you meant to set only one of them. To set only one, open the label, choose **Edit**, remove the other setting under **Configuration Entities**, and then choose **Apply**. Turning the other setting off isn't enough, because the label then turns it off on the devices.

  - Applications in configuration labels lost their icon when you renamed the icon. Renaming an icon now also updates the applications in labels. For an icon that you renamed before the upgrade, select the icon again in the application of the label.
  - A configuration label with background pictures can hold pictures of the base configuration in the picture fields that you didn't set, for example **Additional picture**, and these pictures replace the pictures of the device configuration on the devices. Remove these pictures from the label.

- A configuration label can no longer set the kiosk lock password; devices use the password of their OU. A kiosk lock password can have at most 217 bytes in UTF-8 encoding.
- New installations and upgrades turn on the client check of the Database Layer. A UniconOS Management Board on another computer must be registered on the Scout server, see Configuring the Database Layer client check. After a silent upgrade of such a UniconOS Management Board, only its setup log names the key to register.
- Several Scout Servers on one Scout database: keep UniconOS Management Board and the Scout Board Database Layer running on every server, stop Scout Server on all servers, and then upgrade them one after the other, see Several Scout Servers on one database.
- An administrator can no longer grant rights that they don't hold, so some administrators may be refused actions that worked before. For details, see Administrator rights in UniconOS Management Board.
- UniconOS Management Board now checks the rights of an administrator on each device and configuration that an action reaches. A command to a Dynamic Device Group is refused if the group contains a device on which the administrator doesn't have the right for this command. Changing the predefined image files, update paths or UEFI files needs the matching right **Administration of predefined IDF**, **Administration of predefined update paths** or **Administration of predefined UEFI files**, see Rights on the devices and configurations that an action reaches.
- UniconOS Management Board refuses input that it accepted before:
  - A log folder that isn't a full path on a drive or a network share, for example a relative path or a path with `..`, and a log folder path longer than 226 characters. A log folder that is already set stays in use, so check it after the upgrade.
  - An image file name with characters other than letters, digits and `. _ + ~ -`, that doesn't start with a letter, a digit or `_`, or that is longer than 127 bytes.
  - More than 1,000 USB rules, and a USB rule longer than 1,024 characters.
  - Oversized certificate or picture content, and content of the wrong type.

  - A file destination with a backslash, or one that names a folder instead of a file, in **Files** entries of the advanced device configuration, in configuration labels, in imports and through the REST API. UniconOS Management Board removes spaces at the start or end of a destination, and it adds the file name to a destination that ends with `/`. Entries saved by earlier versions stay unchanged, and the **Files** list marks the entries that devices can't write.
  - An INI file entry whose value contains a line break, and an RDP server or an AVD feed URL with a space inside.
  - REST API calls with an unknown OU path. They now get HTTP status 400.
  - REST API device lookups and searches with a name in `properties` that isn't a device field. They now get HTTP status 400 that names it; earlier versions left the name out of the answer.

- Scout no longer sends **Files** entries of the advanced device configuration whose destination is `/setup/terminal.ini` or `/setup/user.ini`, and it ignores yes/no values in a configuration that it can't read. Devices delete a `/setup/user.ini` that they got this way with their next configuration update and keep their `/setup/terminal.ini`. If devices got settings such as Boot to VDI from such a file, set them in the device configuration or as advanced file entries before the upgrade.
- The **Kiosk Mode** section of the device configuration is removed, because UniconOS devices never applied its settings. Scout no longer sends `/setup/kioskconfig.ini`, devices delete the file with their next new configuration, and at its first start after the upgrade, Scout Server deletes the pass-through passwords that the section stored. This can't be undone. Scripts of your own that read `/setup/kioskconfig.ini` on the devices no longer get it. When you import a configuration exported from an earlier version, UniconOS Management Board skips its kiosk mode settings. To set up kiosk mode, use a copy of a kiosk configuration template, see Configuration templates.
- After the upgrade, Scout sends **Prevent closing the virtual keyboard** to the devices, turned off until you turn it on in UniconOS Management Board. On devices with UniconOS 2611 where a user turned on **Hide keyboard close button**, the first configuration update can turn it off again, and configurations that lock **Mouse/Keyboard** also lock the new user right.
- Scout now sends the PPD model and the printer options in every printer section, so devices with printers get one configuration update after the upgrade.
- The upgrade sets the desktop language, the time zone and the keyboard layout in every device configuration, in the base configuration and in the configurations of OUs and devices, to values that setup takes from the Windows language settings of the Scout server: German, Europe/

Berlin and the German keyboard layout on a German (Germany) server, otherwise English (United States), US/Eastern and the US keyboard layout. Configuration labels keep their values. Note these settings before the upgrade, and set them again in UniconOS Management Board right after the upgrade, before the devices get their next configuration.

- If you set `JWT_SECRET` for UniconOS Management Board yourself, for example in `.env.custom`, give it at least 32 random characters before the upgrade. With a shorter or repetitive secret, setup stops the upgrade before it changes anything, see Session secret of UniconOS Management Board.
- UniconOS Management Board doesn't start if `ZERO_MQ_IS_CURVE_ACTIVATED` is set to `false`, for example in `.env.custom`. Remove this setting before the upgrade.
- After the upgrade, every administrator signs in to UniconOS Management Board again. UniconOS Management Board now ends a session after 30 minutes without activity and 12 hours after the sign-in at the latest, and REST API tokens are valid for 4 hours. If `.env` holds `JWT_EXPIRATION=4h`, the value that earlier versions wrote, setup changes it to `30m`; other values are kept.
- An update now keeps the sign-in domains in `DOMAINS` of UniconOS Management Board. If the domain of the UniconOS Management Board server isn't in the list, the update adds it at the end. If the server moved to another domain, remove its old domain from the list. If an earlier update reset the list to the domain of the server, enter your domains again once.
- With Microsoft Entra ID sign-in through OpenID Connect, add the optional claim `auth_time` to the app registration of UniconOS Management Board. Without it, UniconOS Management Board refuses to grant Software rights, see Configuring authentication options for UniconOS Management Board. SAML sign-in needs no change.
- Standard Windows users can no longer read or change the configuration files of Scout, for example `eluxd.ini` and `install.ini`. Open them in an editor that runs as administrator.
- In a silent upgrade, the service account and database credentials that you pass on the command line now take precedence over the ones that setup saved. Upgrade scripts that still pass an old password apply it, and the Scout services can then fail to sign in.
- With Windows authentication, setup takes the service account password from `install.ini` only if `install.ini` names the account that the Scout Server service runs as. Otherwise, for example after you changed the account of the service in Windows, an update and a change of the Database Layer client check run only silently, with `/v"UCPROP_SERVICEUSER=<account>"` and `/v"UCPROP_SERVICEPASSWORD=<password>"`, and the setup wizard stops with a message.

- Scout Server now refuses a License Server certificate that Windows doesn't trust and that isn't the one accepted when the License Server was configured. If the certificate of your License Server has changed since then, configure the License Server again in UniconOS Management Board after the upgrade, see Licensing.
- Remove Scout Console from every other computer before you upgrade. The upgrade removes the tools only on the computer where it runs, and an earlier Scout Console on another computer can still connect to the upgraded Scout database after a warning about its version. Scout Console isn't tested with UniconOS Management 2611.
- If you run UniconOS Management 2508 LTSR CU2 with Scout Console installed, upgrade directly to 2611. An in-place upgrade to 2607 or 2609 fails with Windows Installer error 2771 ("Upgrade of feature Console has a missing component") and is rolled back, see Upgrading from 2508 LTSR CU2.

Migration from eLux RP 6 to UniconOS

The migration from eLux RP 6 to UniconOS can be carried out easily via a firmware update. For details on the procedure and the system requirements, please refer to the chapter Performing the migration in the product documentation.

Please note the lifecycle milestones of eLux RP 6 so that you can plan the changeover to UniconOS accordingly.

- 2025-03-31 EOM (End of Maintenance eLux RP 6)
- 2025-09-30 EOL (End of Life eLux RP 6)

Update UniconOS images

The firmware update for UniconOS 2611 is carried out as usual via UniconOS Management or via the device. The UniconOS software container required for this must be created via the AllPackages container archive. With UniconOS Management 2611, import the AllPackages container archive into a container in **Software** in UniconOS Management Board, see Adding packages.

From UniconOS 2611, the AllPackages container archive also contains the USB recovery image (`.dd` file). **Software** in UniconOS Management Board and ELIAS 18 import it with the packages, so you can export images for USB recovery sticks without a separate download.

If Scout uses a Citrix License Server, upgrade Scout to UniconOS Management 2611 before you update the devices to UniconOS 2611. UniconOS Management 2508 LTSR to 2609.1 can give the devices an end date of their software maintenance that no longer moves. These devices can

refuse the UniconOS 2611 firmware, and their update log shows "need subscription". For details, see Upgrading Scout before the devices.

Improved certificate verification for firmware updates via HTTPS protocol

Please note that for firmware updates via the HTTPS protocol, the required root/intermediate certificates of a CA (Chain of Trust) must be available on the devices in /setup/cacerts/ in order to be able to perform the required certificate verification. Self-signed certificates can also be used instead of CA certificates.

Disk encryption

Disk encryption is activated by default in UniconOS and can be deactivated/activated via the encryption FPM of the BaseOS package or via a configuration parameter. For further information, see the relevant product documentation topic.

AVD client: new Microsoft Entra application

From UniconOS 2611 on, the Azure Virtual Desktop (AVD) client and the Entra ID sign-in on the device use a new Microsoft Entra application, with the application ID `491bb6b0-7940-4e0f-bb52-b2989db8680b`. An AVD or Entra ID setup that works today can stop working after the update: sign-ins can fail, and Conditional Access policies, user assignments and allowlists that refer to the old application stop applying without any warning. Before you update any device, have an administrator of your Microsoft Entra tenant grant admin consent to the new application and add it wherever the old application is used. Then update the devices of a pilot group, and check that their users can sign in and start their resources. After the update, users sign in once more, and UniconOS creates a new home directory for users who sign in to the device with Entra ID. For the steps, see Moving the AVD client to the new Entra application.

MAC address passthrough with the Lenovo 40AF dock

From UniconOS 2611 on, MAC address passthrough for Lenovo laptops on the ThinkPad Hybrid USB-C with USB-A Dock (type 40AF) needs the MAPT Implementations package with its feature package MAPT for Lenovo clients in the image. Before, it worked without additional packages. Add the package to the image before you update these devices. For details, see Using MAC address passthrough.

Imprivata hot key and inactivity time

From UniconOS 2611 on, devices with Imprivata authentication use the hot key, the inactivity time and the inactivity warning of the Imprivata policy. The policy also decides whether a tap-out, the hot key and inactivity lock the device or log the user off. While a user is logged in, the inactivity time of

the policy replaces the screen saver time of the device's power management settings. The hot key of the policy is reserved for the whole device: applications, also in Citrix sessions, no longer receive it. Check these settings in your Imprivata policies before you update the devices. For details, see Lock or Log Off When Users Leave the Device.

Imprivata XenDesktop automation

From UniconOS 2611 on, Imprivata XenDesktop automation works only on devices with the start scenario Boot to VDI. The device reads each user's desktops from the StoreFront™ stores of the Imprivata computer policy, and users with several desktops pick one in a desktop chooser. The desktops and applications that are selected in a XenDesktop user policy no longer start. On devices without Boot to VDI, users of such a policy get the UniconOS desktop without a Citrix session. Before you update these devices, set them up for Boot to VDI, or change the user policy to **Automate access to applications or published desktops** (XenApp® automation), which works as before on devices without Boot to VDI. Fast user switching with a persistent session isn't supported together with XenDesktop automation. For details, see XenDesktop Automation and the Desktop Chooser.

User rights for new features

Note that new features are enabled for users or administrators unless explicitly stated otherwise. If required, restrict the relevant rights.

Security updates

Further security updates to fix the Intel and AMD processor vulnerabilities are included in the release of UniconOS 2611. The updates consist of patches for the Linux kernel and processor microcode updates for AMD and Intel chipsets. Unicon™ strongly recommends that you perform the relevant system updates for your UniconOS devices. Please note that after updating your devices, appropriate tests in your system environment are necessary to ensure the desired functionality.

Zoom Client for Linux

From UniconOS 2611 on, the AllPackages container archive contains Zoom Client for Linux 7.2.5, which Zoom provides. It replaces version 7.0.0, whose embedded browser components are outdated. If your devices use Zoom, update the Zoom Client for Linux package in their images, and then update the devices.

Client Hardware Compatibility

See Certified Hardware on the Client Hardware Compatibility list for the client models supported by this version of UniconOS.

Deprecation

End of life in UniconOS Management 2611 LTSR

The deprecation of the Scout Console was first announced in December 2025. From UniconOS Management 2611 LTSR, Scout Board is the only management interface of UniconOS Management, and image management is integrated in UniconOS Management Board, see Software.

Scout Console, Scout Report Generator, the legacy ELIAS and the Scout Command Interface (`scmd.exe`) are end of life. The UniconOS Management 2611 LTSR installer neither installs nor offers them. Scout Console and Scout Report Generator are not available for UniconOS Management 2611. UniconOS Management 2609 (15.2609) is the last release that installs these tools. For the device reports of Scout Report Generator, use the device lists of UniconOS Management Board with their filters and the CSV export, see Display devices.

An upgrade to UniconOS Management 2611 LTSR removes these tools:

- Before the upgrade starts, the installer shows a notice that names the tools that it removes and their replacement. A silent upgrade writes the same text to the log file.
- The upgrade deletes `scmd.exe` from the Scout Server folder. A copy that you put back into this folder by hand is deleted again by the next update or repair. Use the UniconOS Management Board REST API instead, see Using the REST API.
- On a computer that has only these tools, the upgrade leaves no Scout installation.

Web-based ELIAS 18 is discontinued with UniconOS Management 2611 LTSR. It was never part of the UniconOS Management installation, and it isn't developed further. To build images, use **Software** in UniconOS Management Board. To take a container over, export it in ELIAS 18, and import it in **Software**, see Importing a container.

From UniconOS 2612, you can build images only with **Software** in UniconOS Management Board. The legacy ELIAS and ELIAS 18 can't build images that contain UniconOS 2612 or later.

If you still use Scout Console, Scout Report Generator, the legacy ELIAS or `scmd.exe`, move these tasks to UniconOS Management Board before you upgrade to UniconOS Management 2611 LTSR. For details, see Upgrading to newer versions.

System requirements

UniconOS Management (formerly Scout 15)

- Windows system
- Management Server: Windows Server 2019, 2022, 2025
- Database:
- MS SQL Server 2019, 2022
- MS SQL Server Express
- MS .NET Framework 4.5.1 or later
- Latest GA version of the MS ODBC driver
- 600 MB hard disk space
- Software Management: at least 20 GB free disk space for its data folder
- Citrix® License Server 2607 (build 56200) or later

UniconOS (formerly eLux 7)

Minimum hardware requirements for UniconOS

- Processor: x86, 64-bit capable and powerful (1)
- RAM: 4 GB (2)
- Storage: 16 GB (3)
- Graphics: AMD or Intel chipset
- Displays: 1920 x 1080 (Full HD)
- Network: Ethernet or WLAN
- Connectivity: USB 3.0 or USB 2.0, USB boot support
- Boot mode: UEFI

(1) 2 GHz (Quad Core) or more recommended, less could affect system performance depending on the use case

(2) at least 3.5 GB available for the operating system (3) 8 GB are also supported, but only with limited functionalities

Recommended hardware requirements for UniconOS

- Processor: Modern x86-64 processor with 4 cores or more, 2 GHz or more
- RAM: 16 GB
- Storage: 64 GB
- Graphics: AMD or Intel, Graphics subsystem supporting hardware-accelerated video rendering and multiple displays
- Displays: Dual 4K monitor support
- Network: 1 Gbit Ethernet and Wi-Fi 6 or later
- Security: TPM 2.0 and Secure Boot
- Connectivity: USB-C and USB 3.x, USB boot support
- Boot mode: UEFI

> **Please note**

> UniconOS has been validated on a variety of hardware configurations. However, users should note that entry-level or legacy hardware may not support all features or may have degraded performance. Limitations may result from insufficient CPU capacity, limited memory, slower I/O subsystems, or lack of hardware acceleration.

Minimum UniconOS Management and SCG version for UniconOS

Please use UniconOS Management versions

- 2302 LTSR CU1 and later or
- 2309 and later

and SCG versions

- 1 2209 and later

for the management of UniconOS devices via UniconOS Management Board and SCG.

From UniconOS 2612, you can build images only with **Software** in UniconOS Management Board, see Software.

Client Hardware Compatibility

UniconOS 2611 has been successfully tested on the following client hardware:

Please note that the devices are tested and released based on UEFI support only. Devices with dual boot mode must be configured to UEFI support. For more information about hardware features, see Hardware limitations in the Release Information.

UniconOS has been validated on a variety of hardware configurations. However, users should note that entry-level or legacy hardware may not support all features or may have degraded performance.

Limitations may result from insufficient CPU capacity, limited memory, slower I/O subsystems, or lack of hardware acceleration. If demanding multimedia usage scenarios impair the system performance of older hardware models with minimal system equipment, it may make sense to upgrade the RAM.

Models that have been successfully tested and approved exclusively by the hardware manufacturer are listed separately in the Manufacturer-Certified Devices section. These devices have not been tested by Citrix.

> **Note:**

> Devices can be added to the HCL through either the Manufacturer-Certified or Citrix-Certified certification process. The certification path depends on the desired level of validation, testing, and support.

> Hardware vendors interested in adding a device to the HCL should start with the Manufacturer-Certified process and contact Citrix to obtain the current test plan and required materials.

> Devices that successfully complete the Manufacturer-Certified process are listed in the HCL as Manufacturer-Certified Devices. For enhanced validation and ongoing compatibility

> testing, Manufacturer-Certified devices may subsequently undergo the Citrix-Certified process.

> Customers interested in a specific device that is not yet listed in the HCL should contact their Citrix representative (ATS) or hardware vendor to discuss certification options.

Citrix-Certified Devices

ACL

All-in-One Medical PCs

- OR-PC21 Medical PC - Intel® Celeron® CPU 3955U @ 2.00GHz

Amulet Hotkey

Stationary Devices

- DX1550 - Intel® Processor U300
- DX1554 - Intel® Processor U300
- DX1714 - 13th Gen Intel® Core™ i5-1335U

Dell

Mobile Devices

- Pro 14 Plus PB14250 - Intel® Core™ Ultra 5 235U vPro
- Latitude 5550 - 13th Gen Intel® Core™ i5-1335U
- Latitude 5540
- Latitude 5440
- Latitude 3440
- Latitude 5530
- Latitude 5520
- Latitude 5420
- Latitude 3420
- Latitude 3410

- Wyse 5470 Mobile Thin Client

Docking Stations

- Pro Thunderbolt 4 Smart Dock – SD25TB4
- Dock WD19
- Dock integrated in Monitor C2422HE
- Dock integrated in Monitor C2722DE
- Dock integrated in Monitor C3422WE

Stationary Devices

- Pro Micro TC Q9M1260 - Intel® Processor N250
- Pro Max Micro FCM2250 - Intel® Core™ Ultra 7 265
- Pro Micro Plus QBM1250 - Intel® Core™ Ultra 5 235T
- OptiPlex Micro Form Factor Plus 7020
- OptiPlex 3000 - Intel® Celeron® CPU N5105
- OptiPlex 3000 - Intel® Pentium® CPU N6005
- OptiPlex 3090 Ultra
- Wyse 5070
- Wyse 7020
- Wyse Z50QQ - up to four monitors
- Wyse 5060
- Wyse 5020

EXTRA Computer GmbH

Mobile Devices

- exone Thin Client Mobile X14 - Intel® N200

Stationary Devices

- exone Thin Client Entry X14 - Intel® N100

Fujitsu

Mobile Devices

- LIFEBOOK E5512 - Intel® Celeron® 7305 @ 1.10GHz
- LIFEBOOK E5511 - Intel® Core™ i7-1165G7 @ 2.80GHz
- LIFEBOOK E5510 - Intel® Core™ i5-10210U CPU @ 1.60GHz
- LIFEBOOK E5410 - Intel® Celeron® CPU 5305U @ 2.30GHz
- LIFEBOOK E559 - Intel® Core™ i5-8265U CPU @ 1.60GHz
- LIFEBOOK E556 - Intel® Celeron® CPU 3855U @ 1.60GHz

Docking Stations

- Thunderbolt ™ 4 Port Replicator
- Dock integrated in Display P34-9 UE - for LIFEBOOK E5410
- Dock integrated in Display P27-9 TE QHD - for LIFEBOOK E5410, FUTRO S7011
- Dock integrated in Display P2410 TS CAM - for LIFEBOOK E5410, FUTRO S7011, FUTRO S9011

Stationary Devices

- ESPRIMO P5010n - Intel® Pentium® Gold G6400 CPU @ 4.00GHz
- FUTRO S9011 with NVIDIA Quadro T400
- FUTRO S7011
- FUTRO S5011
- FUTRO S9010
- FUTRO S9010 with NVIDIA Quadro T400
- FUTRO S9010 with NVIDIA Quadro P400
- FUTRO S7010
- FUTRO S5010
- FUTRO S940
- FUTRO S940 with NVIDIA Quadro P400 - up to five monitors
- FUTRO S740
- FUTRO S540

- FUTRO Q940
- FUTRO S930
- FUTRO S920
- FUTRO S920 with AMD Firepro W2100 - up to four monitors
- FUTRO S720
- ESPRIMO D7010 - Intel® Core™ i3-10100 CPU @ 3.60GHz
- ESPRIMO Q558 - Intel® Core™ i5-8400T CPU @ 1.70GHz

HP

Mobile Devices

- Chromebook x360 14a-ca0506TU - Intel® Celeron® N4120 - only certified for UniconOS 2611
- mt645 G8
- mt645 G7
- Pro mt440 G3
- Elite Mini 600 G9 - Intel® Core™ i5-13500T
- mt46
- mt45
- mt22
- mt21
- EliteBook 850 G6
- EliteBook x360 830 G6

Docking Stations

- 2013 UltraSlim Docking Station - for mt45
- Elite Mini 600 G9
- Elite USB-C Dock G5 - MAC address passthrough for Lenovo ThinkPad L15 Gen4 AMD, L16 Gen1 AMD
- Elite USB-C Dock G4 - for mt21, mt22

Stationary Devices

- Elite t755

- Elite t655
- Pro t550
- t740
- t740 with AMD Radeon E9173 PCIe
- t640
- t540
- t730
- t730 with AMD Firepro W2100 - up to six monitors
- t630
- t530
- t430

IGEL

Stationary Devices

- UD3 M350C

LENOVO

Mobile Devices

- ThinkPad L14 Gen 2 - Intel® Core™ i3-1115G4
- ThinkPad L15 Gen 2 - Intel® Core™ i3-1115G4
- ThinkPad E14 Gen 7 - Intel® Core™ Ultra 5 235U vPro®
- ThinkPad E14 Gen 7 - Intel® Core™ Ultra 5 225U
- ThinkPad L16 Gen1 - Intel® Core™ Ultra 5 125U
- ThinkPad L14 Gen5 - Intel® Core™ Ultra 5 125U
- ThinkPad L13 Gen5 - Intel® Core™ Ultra 5 125U
- ThinkPad L14 Gen4 - Intel® Core™ i5-1345U
- ThinkPad L15 Gen4 - Intel® Core™ i3-1315U
- ThinkPad L14 Gen4 - Intel® Core™ i3-1315U
- ThinkPad L13 Gen4 - Intel® Core™ i3-1315U
- ThinkPad L14 Gen6 - AMD Ryzen 5 Pro 215 with Radeon 740M Graphics

- ThinkPad L16 Gen1 - AMD Ryzen 3 PRO 7335U with Radeon Graphics
- ThinkPad L14 Gen5 - AMD Ryzen 5 PRO 7535U with Radeon Graphics
- ThinkPad L15 Gen4 - AMD Ryzen 3 PRO 7330U
- ThinkPad L14 Gen4 - AMD Ryzen 3 PRO 7330U
- ThinkPad L13 Gen4 - AMD Ryzen 3 PRO 7330U
- ThinkPad L14 Gen1 - AMD Ryzen 5 PRO 4650U

Docking Stations

- ThinkPad Universal Thunderbolt 4 Dock 40B0
- ThinkPad Universal USB-C Dock
- ThinkPad Hybrid USB-C with USB-A Dock TYPE 40AF - for ThinkPad L14 Gen4, Dell Latitude 3410 - MAC address passthrough for Lenovo laptops needs the MAPT Implementations package with its feature package MAPT for Lenovo clients

Stationary Devices

- ThinkCentre M90n-1 - Intel® Celeron® CPU 4205U @ 1.80GHz
- ThinkCentre M720q Tiny - Intel® Core™ i5-9500T
- ThinkCentre M710q 10 MS - only certified for UniconOS 2601
- ThinkCentre Neo 50q Gen 4 - Intel® Core™ i3
- ThinkCentre Neo 50q Gen 4 - Intel® Celeron® 7305
- ThinkCentre M70q (Gen 5) - Intel® Core™ i3-14100T
- ThinkCentre M70q (Gen 3)
- ThinkCentre M70q (Gen 2) - Intel® Pentium® Gold G6405T CPU @ 3.50GHz
- ThinkCentre M75q (Gen 5) - AMD Ryzen 3 PRO 8300GE
- ThinkCentre M75q (Gen2)
- ThinkCentre M75n
- ThinkCentre M715q
- ThinkCentre M625q - AMD E2-9000e
- ThinkCentre M625q - AMD A4-9120e
- ThinkStation P3 Tiny - with Lenovo ThinkCentre TIO22 (Gen 5) Touch / TIO22 (Gen 5) Touch / TIO27 LCD Monitor

LG

Mobile Devices

- gram Mobile Thin Client 17ZT90R-G.AX34G - 13th Gen Intel® Core™ i3-1315U
- gram Mobile Thin Client 15ZT90R-G.AX34G - 13th Gen Intel® Core™ i3-1315U
- gram Mobile Thin Client 17ZT90P-G.AX36G
- gram Mobile Thin Client 15ZT90P-G.AVD11KB
- Ultrabook 14UT70Q-G.AX34G

Stationary Devices

- CQ601N-2P Thin Client Box - Intel® Pentium® Silver N6005
- Thin Client CL600N-6N

All-in-One Devices

- Medical All-in-One Cloud Device 24CR671N-6P (24”)
- Medical All-in-One Cloud Device 24CN670N-6N (24”)
- All-in-One Cloud Device 24CQ651N-2P (24”) - Intel® Pentium® Silver N6005

MOStron

Panel Devices

- Touch-Panel-PC TSPC5-24

NEXterminal

Mobile Devices

- NEXlaptop IV
- NEXstation XI DP/DP2 - Intel® Quad Core 2.0Ghz
- NEXstation X

Manufacturer-Certified Devices

Asus

Stationary Devices

- NUC 14 Essential Slim PC Kit RNUC14MNK9700002 - Intel® Processor N97

Centerm

Stationary Devices

- S25 Ultra Mini PC - Intel® Processor N150

Dell

Docking Stations

- Dock WD19S

Fujitsu

Mobile Devices

- LIFEBOOK E5512 / MTC - Intel® Core™ i5-1235U up to 4.40 GHz, identical to LIFEBOOK E5512
- LIFEBOOK E5512 / MTC - Intel® Core™ i3-1215U up to 4.40 GHz, identical to LIFEBOOK E5512
- LIFEBOOK E5412 / MTC - Intel® Core™ i3-1215U up to 4.40 GHz, identical to LIFEBOOK E5512
- LIFEBOOK E5412 / MTC - Intel® Celeron® 7305 @ 1.10GHz, identical to LIFEBOOK E5512
- LIFEBOOK E5510 / MTC - Intel® Core™ i3-10110U up to 4.1GHz, identical to LIFEBOOK E5410
- LIFEBOOK E5510 / MTC - Intel® Celeron® CPU 5305U up to 2.30GHz, identical to LIFEBOOK E5410
- LIFEBOOK E5410 / MTC - Intel® Core™ i3-10110U up to 4.1GHz, identical to LIFEBOOK E5410

Docking Stations

- Thunderbolt ™ 4 Port Replicator - for LIFEBOOK E5x12
- Dock integrated in Display P34-9 US - for LIFEBOOK E5x10, FUTRO S7011
- Dock integrated in Display P27-9 TE QHD - for LIFEBOOK E5x10
- Dock integrated in Display P2410 WE CAM - for LIFEBOOK E5x10, FUTRO S7011
- Dock integrated in Display P2410 TS CAM - for LIFEBOOK E5x10

Lenovo

Mobile Devices

- ThinkPad T14 Gen7 - AMD Ryzen 5 Pro 215 with Radeon 740M Graphics

Stationary Devices

- V100q - Intel® N100

ThinClient Direct

Mobile Devices

- ThinClient Direct TCD M (001T) - Intel N150

Stationary Devices

- ThinClient Direct TCD 1 (001) - Intel N5105
- ThinClient Direct TCD 1 (002) - Intel N200
- ThinClient Direct TCD 2 (001) - Intel N5105
- ThinClient Direct TCD AIO (001) - Intel N5105
- ThinClient Direct TCD G (001) - Intel N150
- ThinClient Direct TCD I (001) - Intel N150

UltrArmor

Stationary Devices

- UA3 N370 - AMD Ryzen Embedded R2312
- UA3 N370 Plus - AMD Ryzen Embedded R2314

General information

Administrator's Guides describe all features and functionality of a product or component in detail. They can be used as a reference manual. For some procedures and best practices, we provide **Short guides** to help you perform particular tasks. **Whitepapers** give an overview of benefits and services, standards or technical topics.

This documentation refers to the latest product versions. Any features added or changed for a particular release are listed in the Release Notes on `www.myelux.de`.

Representation

REPRESENTATION	DESCRIPTION
Control element	All graphical user interface controls are displayed in **bold**.
[Menu > menu command]	Multiple sequential steps through the interface such as menus or dialog are separated by `>`.
Value	All data that have to be entered by the user or data that represent a field value are displayed in `Courier New`. Also, file names and path names are displayed in `Courier New`.
[Strg]	Keys to be pressed are displayed in [capital letters].
[<Placeholder>]	Placeholders in instructions and user input are displayed in *italics* and in `<angle brackets>`.
1. Instruction	Procedures to be carried out step by step are realized as numbered steps.
Result	System responses and results are displayed in *italics*.

Abbreviations and acronyms

ABBREVIATION	DESCRIPTION
AD	Active Directory, directory service of Microsoft Windows Server
DMZ	Demilitarized zone. Systems in a DMZ are protected against other networks by firewall rules.
EBKGUI	Interface of the eLux® Builder Kit (Tool for creating eLux software packages)
EPM	eLux package module ([.epm], software package)
FPM	Feature package module ([.fpm], part of a software package)
FQDN	Fully qualified domain name. DNS name that uniquely identifies a computer on the network, example: `client1.sampletec-01.com`
GB	Gigabyte
GHz	Gigahertz (processing speed)
HDD	Hard disk drive (flash memory)
IDF	Image Definition File ([.idf])
IIS	Internet Information Services, Microsoft Web server
MB	Megabyte
OU	Organizational unit. Unit or group within the organizational structure
VM	Virtual machine
VPN	Virtual Private Network

Administrator's guides

An administrator\'s guide is a product manual that describes the features and functions of one of our products, including detailed instructions on how to proceed.

We provide administrator's guides for our operating system UniconOS (formerly eLux®), for UniconOS Management Board, the management interface of UniconOS Management (formerly Scout Enterprise Management Suite®), and for additional products such as Scout Cloud Gateway and Scout Agent for Windows.

The following guides describe tools that are end of life. From UniconOS Management 2611 LTSR, these tools are no longer part of the installation:

- **UniconOS Management**, the guide for the Scout Console
- **Scout Report Generator**
- **ELIAS**, the guide for the legacy ELIAS

ELIAS 18 is discontinued with UniconOS Management 2611 LTSR, see Deprecation. To build images, use Software in UniconOS Management Board.

The documentation refers to the latest product versions.

UniconOS

UniconOS (formerly eLux® 7) is a hardware-independent operating system designed for cloud-computing environments.

Overview and general information

About eLux 7

Keyboard shortcuts

Touchpad gestures for mobile devices

About eLux® 7

eLux 7 is a hardware-independent operating system designed for cloud-computing environments. It runs on mobile and stationary thin clients, common PCs, laptops, and any other endpoint with x86 architecture. eLux is based on Linux and provides a write-protected file system, which makes it secure against computer viruses and other malware.

eLux 7 comes with a new desktop that is specially tailored to hybrid work scenarios and a new partition layout that offers more space for installing apps on the end device.

You can find information on migrating existing end devices with eLux RP in the Shortguide migration of eLux RP 6 to eLux 7.

This guide supports the system administrator in setup, maintenance and operation of devices and PCs running eLux 7, hereafter referred to as **eLux** devices. The documentation refers to the current version eLux 7.

This guide assumes knowledge of:

- Installation, maintenance and operation of computer networks and peripherals
- Operating system skills of the server machines in use

> **Note:**

> eLux 7 devices can be centrally managed through the Scout Enterprise Management Suite® version 15.x.

> The management protocol for communication between Scout Server and eLux devices offers end-to-end encryption via TLS 1.2. TLS-encrypted communication is done via port **22125**.

For support periods, see the Whitepaper **Releases, Lifecycles and Compatibility**.

Keyboard shortcuts

These keyboard shortcuts let you control the eLux® desktop without a mouse — switching between applications, locking the device, and opening key panels:

SHORTCUT	FUNCTION
WIN	Open or close the **Start menu**.*

SHORTCUT	FUNCTION
CTRL+ALT+↓	Switch to the next open application.
CTRL+ALT+↑	Switch to the previous open application.
CTRL+WIN	Switch from other applications to eLux: The eLux taskbar/system bar with open applications is shown.
WIN+ALT+C	Open **Device configuration** on the section you used last.
WIN+ALT+I	Open **Device configuration** on **Information**.
CTRL+ALT+D	Switch between the open application and the eLux desktop.
CTRL+ALT+Home	Unlock all dialogs of **Device configuration**: The **Unlock all dialogs** window requests the local device password.
CTRL+ALT+END	Lock the device's screen. If user authentication is active, the user password is required for unlocking. With Imprivata authentication, the Imprivata policy can make this shortcut log the user off instead, see Lock or Log Off When Users Leave the Device.
CTRL+ALT+Function key	Switch between the consoles, if the **Console switch** option is enabled.
	The following consoles are available
	F1: eLux desktop
	F4: Message console

* A quick tap (press and release within about half a second) opens or closes the Start menu. Holding the key, or pressing it together with another key, does nothing. When a full-screen application or a focused remote session (Citrix, RDP, Horizon) has focus, the key goes to that application instead — the shortcut works again once you click back on the desktop.

Touchpad gestures for mobile devices

Mobile devices often need to be operated via a touchpad and do not always have hardware mouse buttons.

By default, mouse actions are mapped to the following touchpad gestures:

MOUSE ACTION	TOUCHPAD GESTURE
Click with left mouse button	Tap one finger on the touchpad (single-tap)
Right-click (right mouse button)	Tap two fingers on the touchpad (single-tap)
Middle-click (wheel button)	Tap three fingers on the touchpad (single-tap)
Move cursor	Drag your finger on the touchpad
Scroll	Slide two fingers at the same time

You can change the scroll direction in the touchpad settings. See Keyboard and Mouse.

In Google Chrome, Chromium, and Firefox, you can also zoom a web page with two fingers, as on a laptop. Spread the fingers apart to zoom in. Pinch them together to zoom out. The gesture doesn't zoom pages in the built-in browser.

Installation

eLux® can be installed directly on a device's flash memory or on a hard disk. The installation procedure is a form of recovery installation and can be performed in two ways:

- From USB stick: for all supported operating system versions, we provide an **eLux USB Stick** image, available for download on our myelux.com portal, for creating an installation stick.
- Via PXE recovery: for large environments, PXE-capable devices can be installed over the network if the eLux software container and Scout Enterprise Management Suite® are already installed.

Both procedures are described in detail in our short guide **eLux Recovery procedures**.

Installation in a VM

eLux can be installed in a VM (Virtual Machine). Depending on the virtualization solution, specific settings need to be activated.

> **Note:**
>
> It is recommended to use at least 16 GB of storage. The eLux ISO image needs to be mapped as a drive for the installation.

XenServer®

SETTING	VALUE
Guest OS	Ubuntu Jammy Jellyfish 22.04 (Ubuntu)
Boot Mode	UEFI Boot

> **Note:**
>
> See also eLux 7 2503 VM on XenServer as a reference.

VMware

Types: vSphere, Fusion, Workstation

SETTING	VALUE
Guest OS	Other (64-bit)
Boot Option - Firmware	(U)EFI

Virtual Box

SETTING	VALUE
Type	Linux
Subtype	Ubuntu
Version	Ubuntu (64-bit)
Extended feature	Enable I/O APIC
Extended feature	Enable EFI

Further information

System requirements

First boot procedure

First configuration

Device password

Self-administration on the device

System requirements

Hardware requirements eLux® 7

	MINIMUM REQUIREMENTS
Processor	x86, 64-bit capable and powerful (2 GHz quad-core or more recommended, less could affect system performance depending on the use case)
RAM	4 GB (at least 3.5 GB available for the operating system)
HDD	16 GB (8 GB are also supported, but only with limited functionalities)
GPU (Graphics processing unit)	AMD or Intel chipset
Network	Ethernet or WLAN
I/O ports	USB 3.0 or USB 2.0, USB boot support
Resolution	1920 x 1080 (Full HD)
Boot mode	UEFI

Certified hardware

The supported hardware models (Hardware Compatibility List) for each operating system version are published on our technical myelux.com portal, within the current eLux container under **Certified Hardware** (including previous versions).

For support periods and the compatibility matrix, see the Whitepaper **Releases, Lifecycles and Compatibility**.

Minimum Scout and Scout Cloud Gateway versions for eLux 7

Use the following versions to manage eLux 7 devices through UniconOS Management Board and Scout Cloud Gateway:

Scout:

- 15 2302 LTSR CU1 and later
- 15 2309 and later

Scout Cloud Gateway:

- 1 2209 and later

To build images, use **Software** in UniconOS Management Board, which comes with UniconOS Management 2611 and later, see Software. Images with UniconOS 2612 or later can be built only with Software.

First boot procedure

eLux® goes through the following first boot procedure when in its initial state, after a factory reset, or after a recovery installation:

a. Scan UEFI

b. Make a DHCP server request

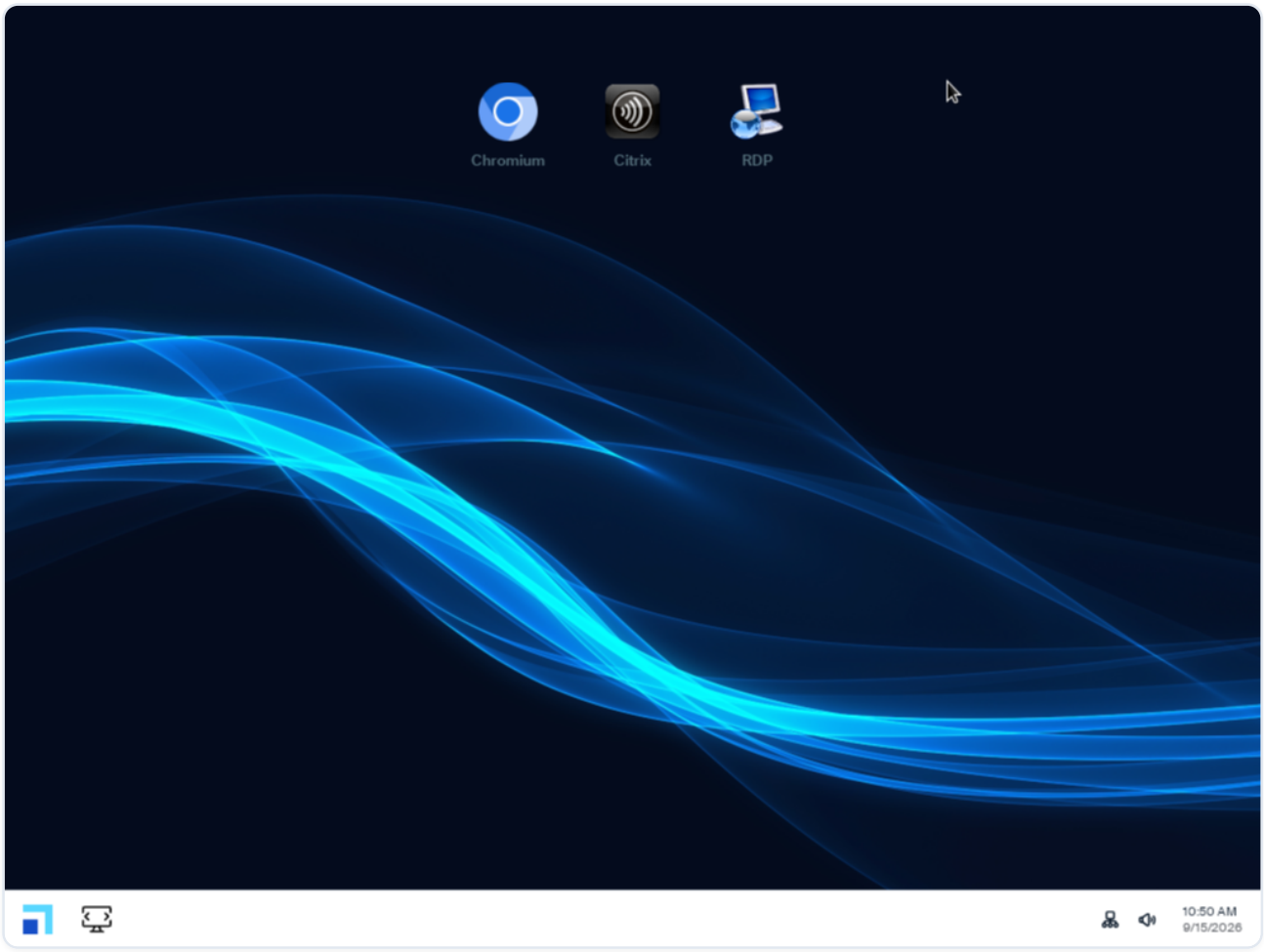
> **Note:**
>
> To enable the device to connect to the Scout Server, either DHCP or DNS must be configured, see Registering Scout Server in the network. For the settings that Scout applies to new devices, for example the OU that they are assigned to, see Devices in the **UniconOS Management Board** guide.

> c. Start the eLux operating system

If either DHCP or DNS has been configured for the Scout Server, the device is automatically registered in the Scout infrastructure and receives a new configuration.

If a device cannot retrieve the IP address of the Scout Server, the First configuration opens and leads you through the first configuration.

Once the first boot procedure, and if applicable the first configuration, is complete, the eLux desktop appears.



First configuration

During the first boot procedure, a wizard opens to guide you through the first configuration. The First Configuration Wizard also starts when you reset the device to factory state.

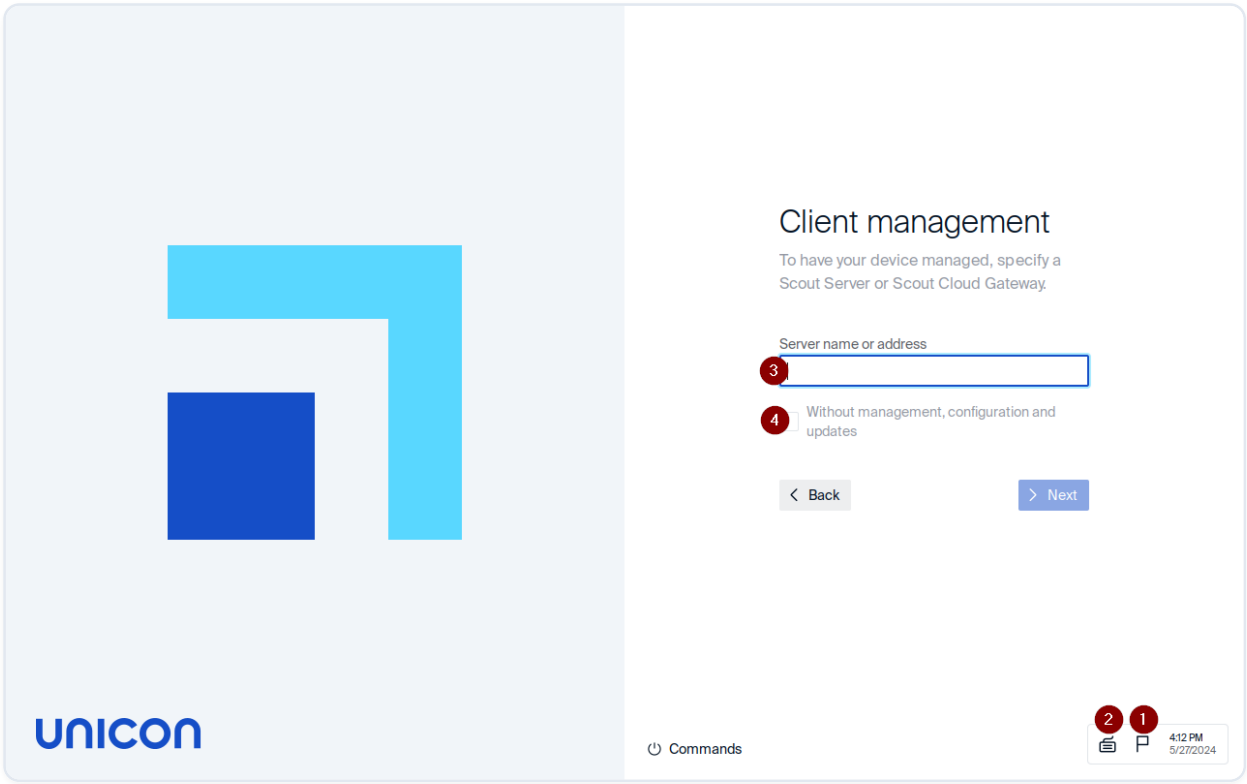
The First Configuration Wizard offers the following options:

- Manage the device through UniconOS Management Board. The configuration data is transferred from the Scout Server.
- Connect the device through the Scout Cloud Gateway and manage it through UniconOS Management Board. The configuration data is transferred from the Scout Server.
- Configure eLux® manually, which means locally on the device.

Going through the first configuration and connecting to a Scout Server

1. Select the display language **(1)** and keyboard language **(2)**.

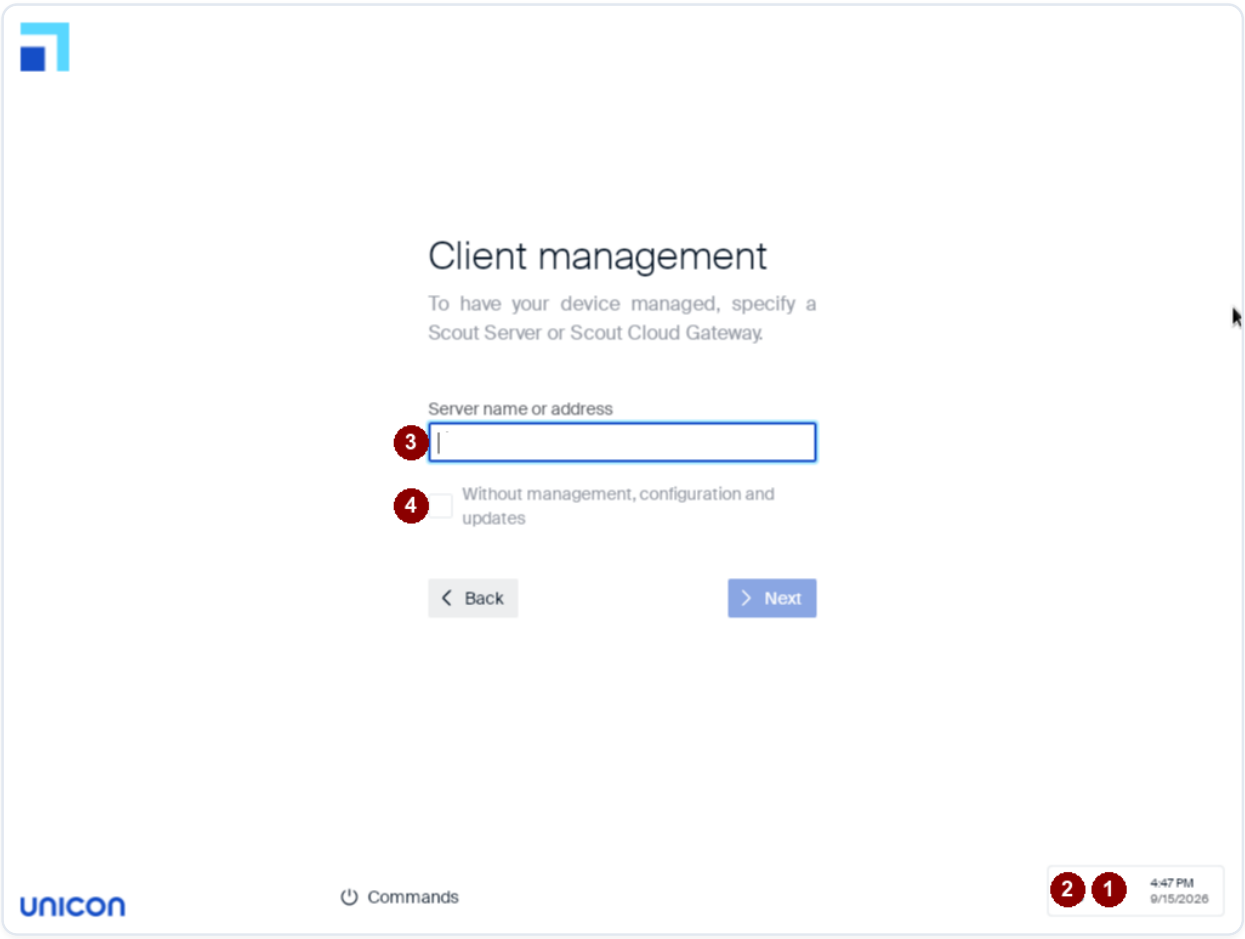
   The following languages are supported: English, German, French, Spanish, Korean, Hebrew, Turkish, and Dutch.



2. Enter the address of the Scout Server as FQDN or IP address *(3)*.

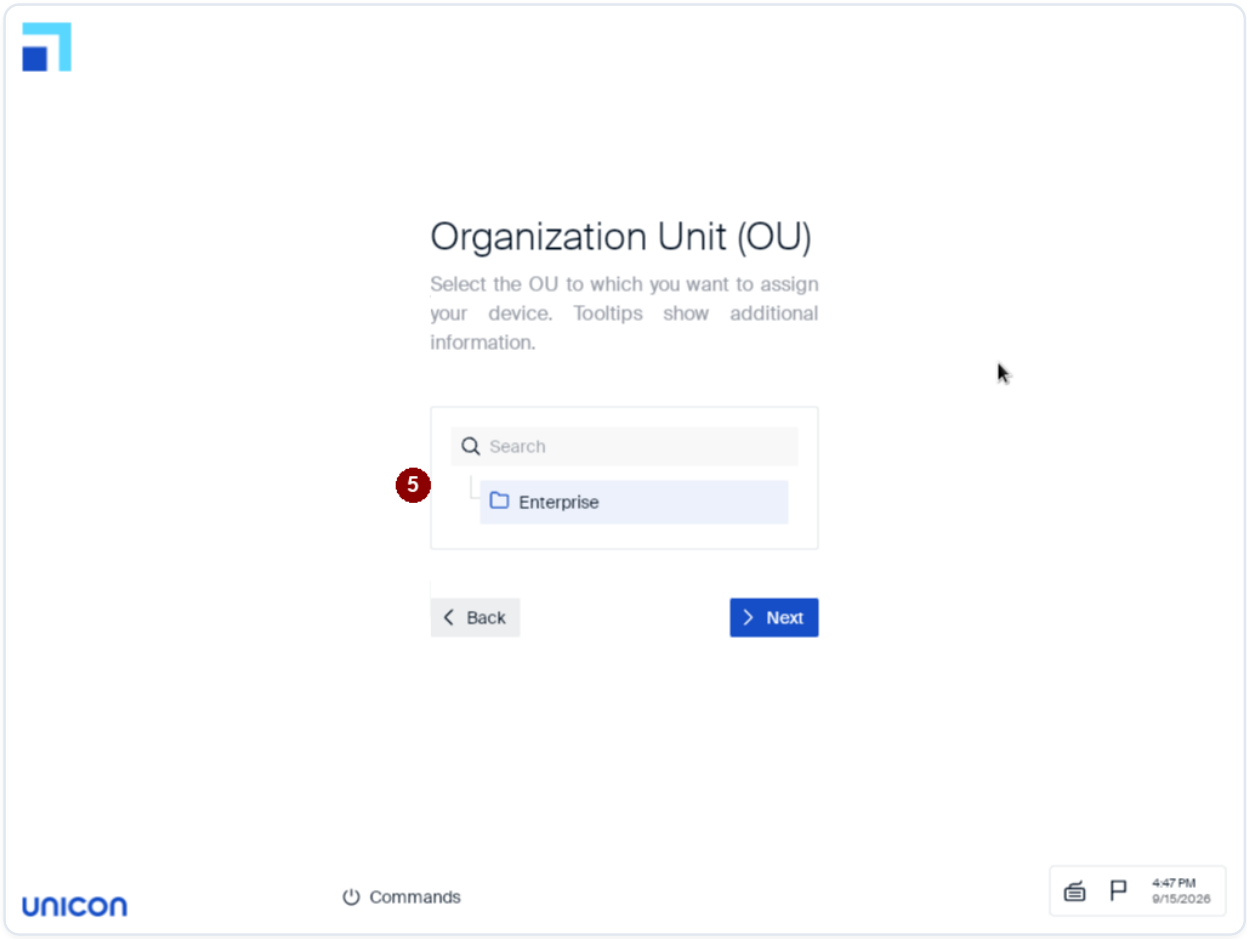
   If your device connects through the Scout Cloud Gateway, enter the address of the Scout Cloud Gateway instead.

   If you don't want the device managed through a Scout Server or Scout Cloud Gateway, select **Without management, configuration and updates** *(4)* instead. In this case, the wizard skips directly to the license agreement, and then to the Summary screen, without the organization unit and device name steps described below.



3. Click **Next**.
4. If you specified a Scout Server or Scout Cloud Gateway, select the destination OU (organization unit) to which the device is added *(5)*.

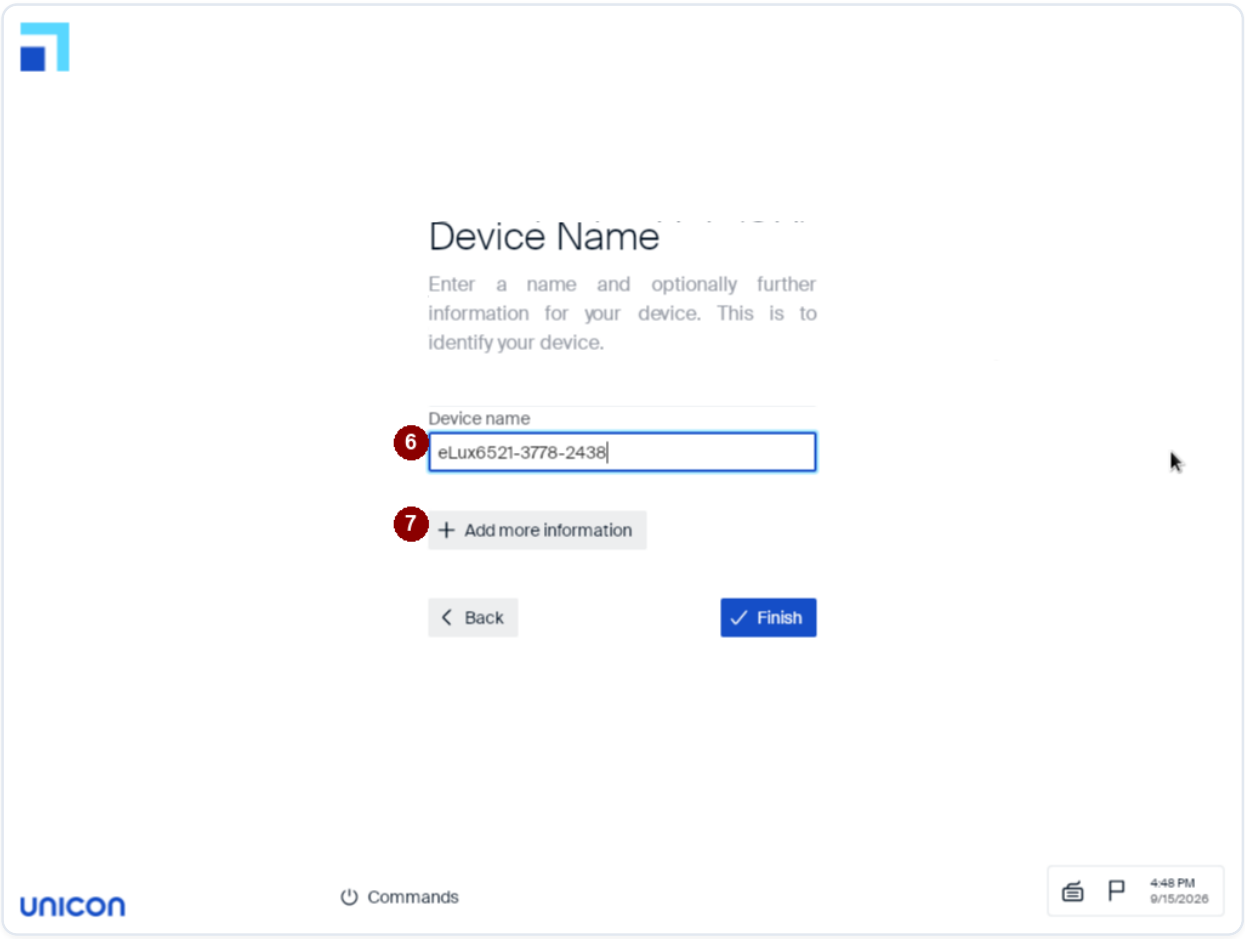
   Use the search box to find the OU. Tooltips show additional information about each entry.



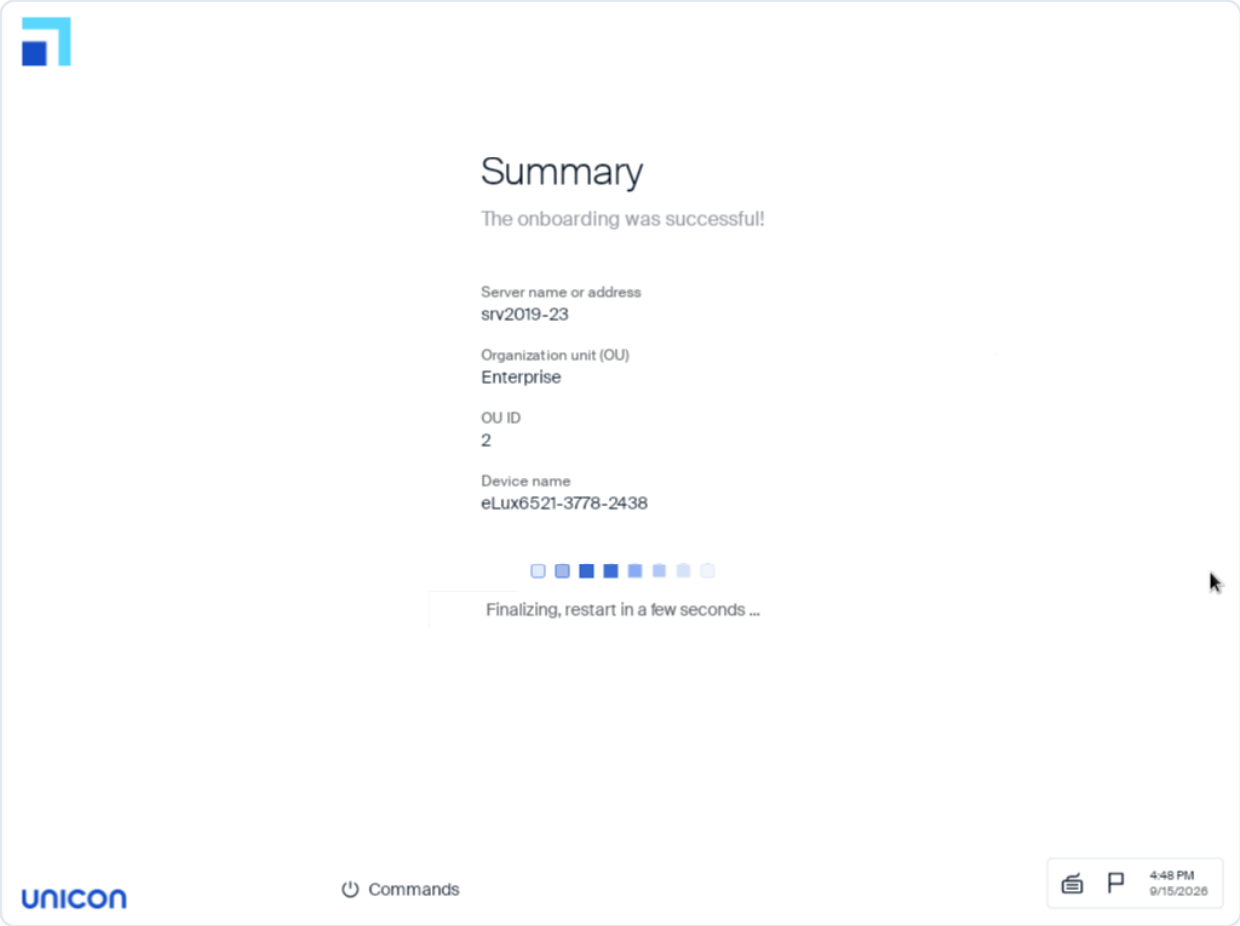
5. Click **Next**.
6. Enter a name for the device *(6)*.

   For the device name format rules, see Device names in the **UniconOS Management Board** guide.

   Optionally, click **+ Add more information** *(7)* to enter further device details.



7. Confirm with **Finish**.
8. Review your entries on the Summary screen.



If you specified a Scout Server or Scout Cloud Gateway, the device is registered in the Scout infrastructure, added to its destination OU, and restarted. The device contacts the Scout Server and downloads the configuration and application data of the destination OU. If you selected **Without management, configuration and updates**, *the device restarts without connecting to a Scout Server or Scout Cloud Gateway. If you installed eLux using a USB recovery stick, remove the stick before the device restarts.*

If the device already exists in UniconOS Management Board, for example because you added it with its MAC address, it gets the configuration that Scout already has for it, see Adding devices.

For further information on connecting through the Scout Cloud Gateway, see Integrating new devices in the Scout Cloud Gateway guide. For further information on managing devices with Scout, see the **UniconOS Management Board** guide.

Device password

All devices managed by a Scout Server receive the same device password. There is only one device password for all devices in the same infrastructure, defined in the base device configuration.

The device password is used for unique assignment and authentication to the Scout Server, so that no other Scout Server can manage this device.

In the initial state, the device password is `elux`.

> **Note:**
>
> The device password has nothing to do with a user password, which is used for example for AD user authentication.

> For further information on passwords, see also Passwords in the **Scout** guide.

Self-administration on the device

With administrator rights, users can change the configuration locally on the device or completely disconnect from the Scout infrastructure. To prevent abuse, we recommend that you change the initial device password and do not release it. For further information, see Local device password in the **Scout** guide.

Logging on with full access on the device

1. Open Device configuration.
2. Press `Ctrl+Alt+Home`. The **Unlock all dialogs** window opens.
3. In the **Password** field, enter the device password, and then click **Apply**.

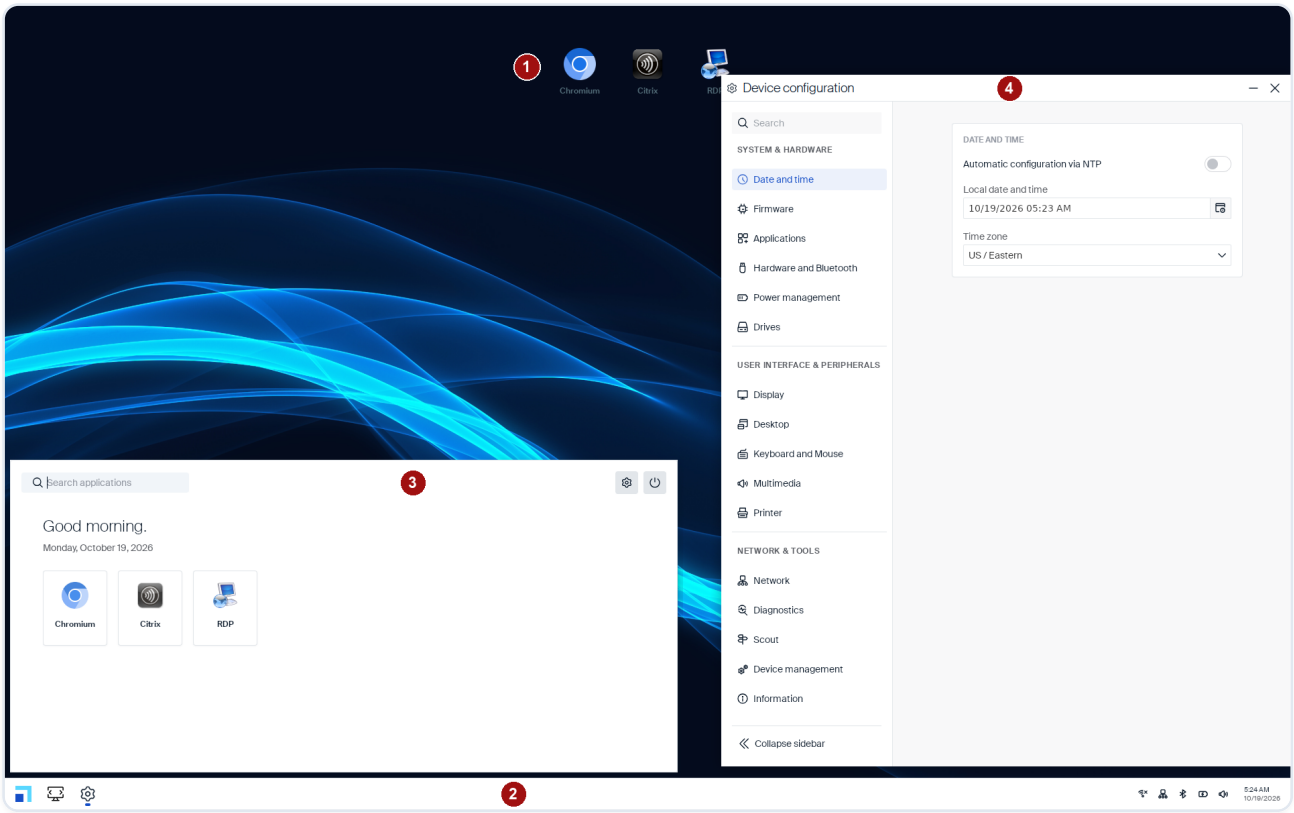
If the password is wrong, the message **The password is incorrect** is displayed.

You have full access rights to all sections of Device configuration on the device.

eLux® 7 user interface

> **Note:**
>
> eLux 7 devices can be centrally managed through the Scout Enterprise Management Suite version 15.x. For further information, see Compatibility client platform and Scout Enterprise Management Suite in our **Releases** Whitepaper.

> The screenshot shows the main elements of the eLux 7 user interface.



The numbers in the screenshot mark these elements:

	DESCRIPTION
1	**Desktop** with applications.
2	**System bar** with the **Start menu** button, running applications and system icons.
3	**Start menu** with application search, the **Device configuration** icon and the **Power** menu.
4	**Device configuration**.

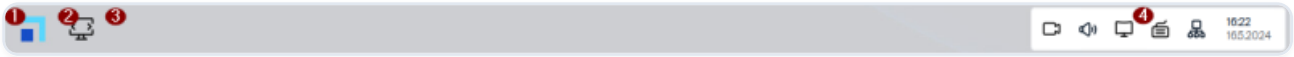
System bar

Device configuration

Applications in the eLux 7 interface

System bar

The **System bar** provides various functions and can be configured.



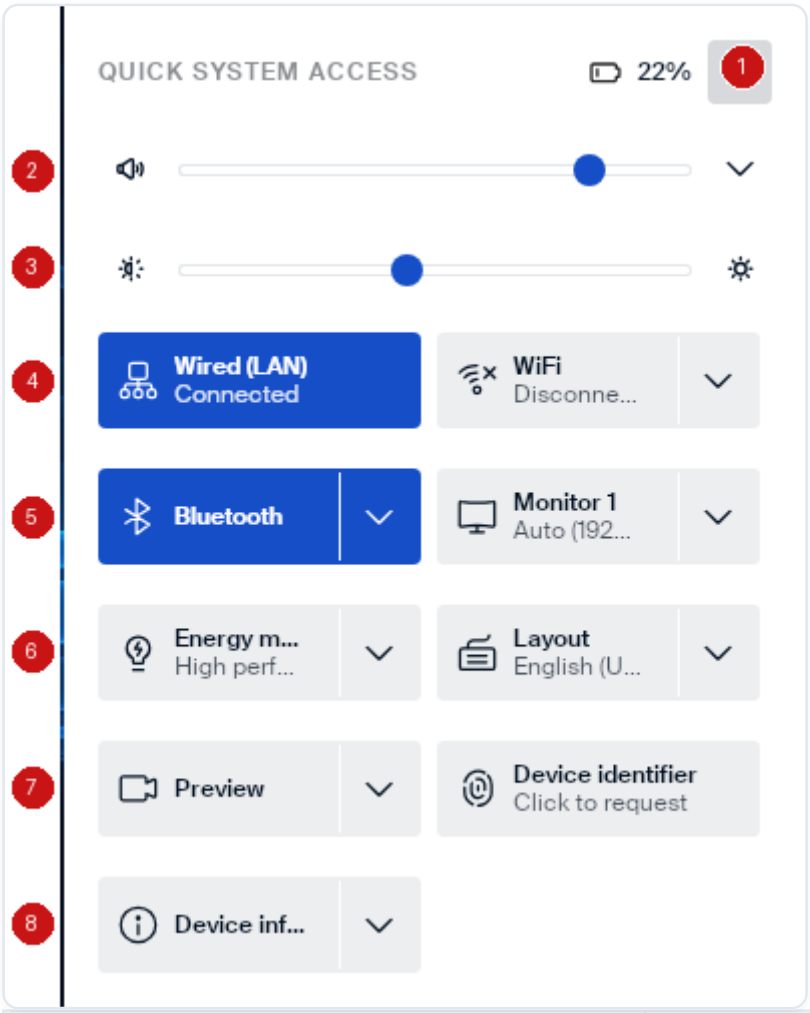
	DESCRIPTION
1	Show **Start menu** with the application selection, **Device configuration** and the power commands (see below)
2	Show desktop
	Minimizes all open windows.
3	Taskbar for displaying applications with an open window.
4	System icons to easily change the settings for hybrid work scenarios (e.g. video, audio, monitor layout, network). Click a system icon to open the **Quick system access** panel. Tooltips display status information for the system icons as soon as the mouse pointer is positioned over the icon. If an administrator allows parallel connections, there is a network icon for each connected network, also on the login screen and the lock screen, see Using Ethernet and Wi-Fi at the same time.
	To display information about active third-party software (e.g. Zoom, Cisco Secure Client), click on the up arrow -icon
	If a USB storage device is connected, its icon appears and can be clicked to open the **USB info** popup (see Viewing USB device info). The icon is not shown otherwise.
	Click the time to open a popup with the local date and time and the time zone. In the popup, click the gear icon to open **Device configuration** on **Date and time**.

Start menu

Notifications

Using Quick system access

Click any system icon (for example audio or network) to open **Quick system access**. It provides direct access to the most common settings without opening the full Device configuration.



	DESCRIPTION
1	Battery charge level. Click the gear icon to open **Device configuration** on **Multimedia**.
2	**Volume** slider. Click the arrow to expand separate sliders for the individual audio devices.
3	**Brightness** slider for the display.
4	**Wired (LAN)** and **WiFi** tiles, showing the connection status. Click a tile to connect, disconnect, or open its connection details. If an administrator allows parallel connections, both tiles can show **Connected** at the same time,

	DESCRIPTION
	see Using Ethernet and Wi-Fi at the same time.
5	**Bluetooth** tile to turn Bluetooth on or off. **Monitor 1** tile, showing the active resolution; click to open display settings.
6	**Energy mode** tile, showing the active power management profile. **Layout** tile, showing the active keyboard layout.
7	**Preview** tile to show or hide the webcam preview. **Device identifier** tile; click **Click to request** to have the device request and display a fresh device identifier from Scout.
8	**Device info** tile. Click to expand a summary of device details such as MAC address and serial number.

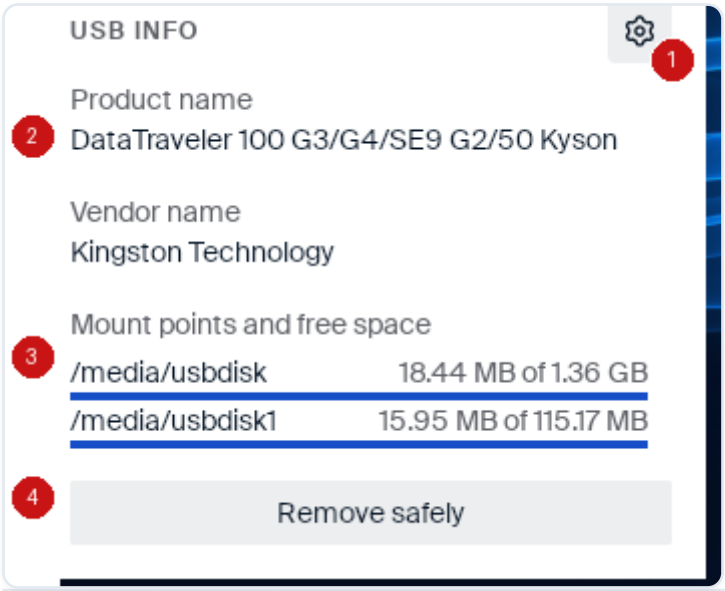
Click a tile's arrow to expand it for more options, or click the tile itself to toggle it, depending on the tile.

Viewing USB device info

> **Note:**
>
> The USB icon and the **USB info** popup are only available while a USB storage device is connected. The icon disappears again after the device is removed.

When a USB storage device is connected, its icon appears in the system bar, as shown below:

Click the icon to open the **USB info** popup:



	DESCRIPTION
1	Click the gear icon to open **Device configuration**, on the **Hardware and Bluetooth** dialog.
2	**Product name** and **Vendor name** of the connected device.
3	**Mount points and free space**, listing each partition's mount point and the free space out of its total capacity.
4	Click **Remove safely** to unmount the device before physically disconnecting it.

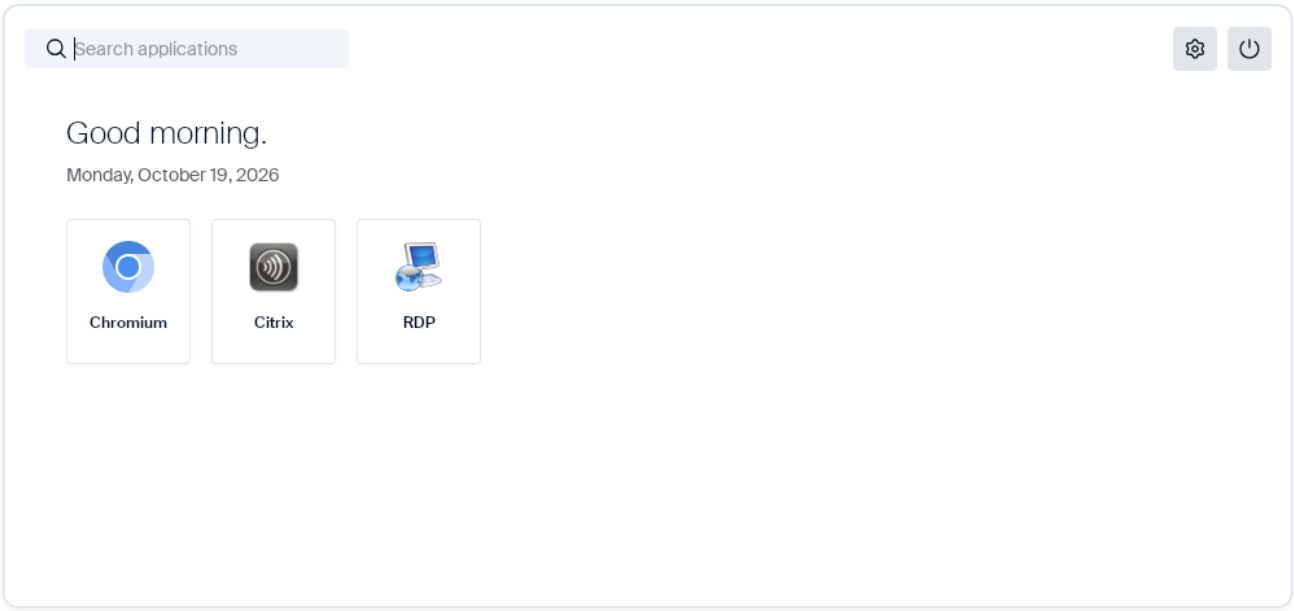
Start menu

The **Start menu** gives you access to your applications, the Device configuration and the power commands.

Which items are displayed depends on whether the device is managed by Scout, whether user authentication is configured, and the user rights the administrator has defined.

- To show the Start menu, on the left of the system bar, click the eLux icon .

START MENU ELEMENTS



The Start menu has these elements:

ELEMENT	DESCRIPTION
Search field	Searches all applications, including the applications of a Citrix® Store.
Application icons	List all predefined applications.
Device configuration	Opens the device configuration, which provides the settings for the eLux® Client.
Power	Opens a menu with the power commands.

To display the applications of a Citrix Store, left-click the Citrix Store icon.

For more information about the device configuration, see Device configuration.

The **Power** menu has these commands:

COMMAND	DESCRIPTION
Lock	Locks the screen. To unlock it, the user authenticates again, for example with their password. On a kiosk device with the kiosk lock screen, the user enters the kiosk password.

COMMAND	DESCRIPTION
Log off	Logs the user off and shows the login screen.
Sleep	The device is set to standby mode (Suspend to RAM).
Restart	The device is shut down and restarted immediately.
Shut down	The device is shut down and switched off.

The commands to update the firmware, fetch the configuration, change credentials and reset the device are in **Device management**. For more information, see Device management.

Notifications

System notifications inform users about a changed network status or the connection or removal of peripheral devices, for example. Notifications are shown in a small message window at the bottom right of the screen and are removed after a few seconds.

To make it easier for users to read several messages, especially if they arrive at short intervals, an additional icon allows you to show all notifications in a separate panel. This panel displays the notifications one under the other and remains active until you close it with a click.

1	To open the Notification Center, click the temporarily displayed icon.
2	Notifications are displayed in the Notification Center according to their display duration.
	System notifications from hardware detection disappear automatically after a few seconds.
	The display duration for messages from the Scout administration is set in the **Send message** command in UniconOS Management Board, see Commands and schedules.

	Notifications are listed one below the other if several are waiting. To close a notification, click its close field.
	The panel with further notifications remains open.

Device configuration

Device configuration is where you set up the device, for example its applications, hardware, network and desktop. Each topic has its own section.

Opening Device configuration

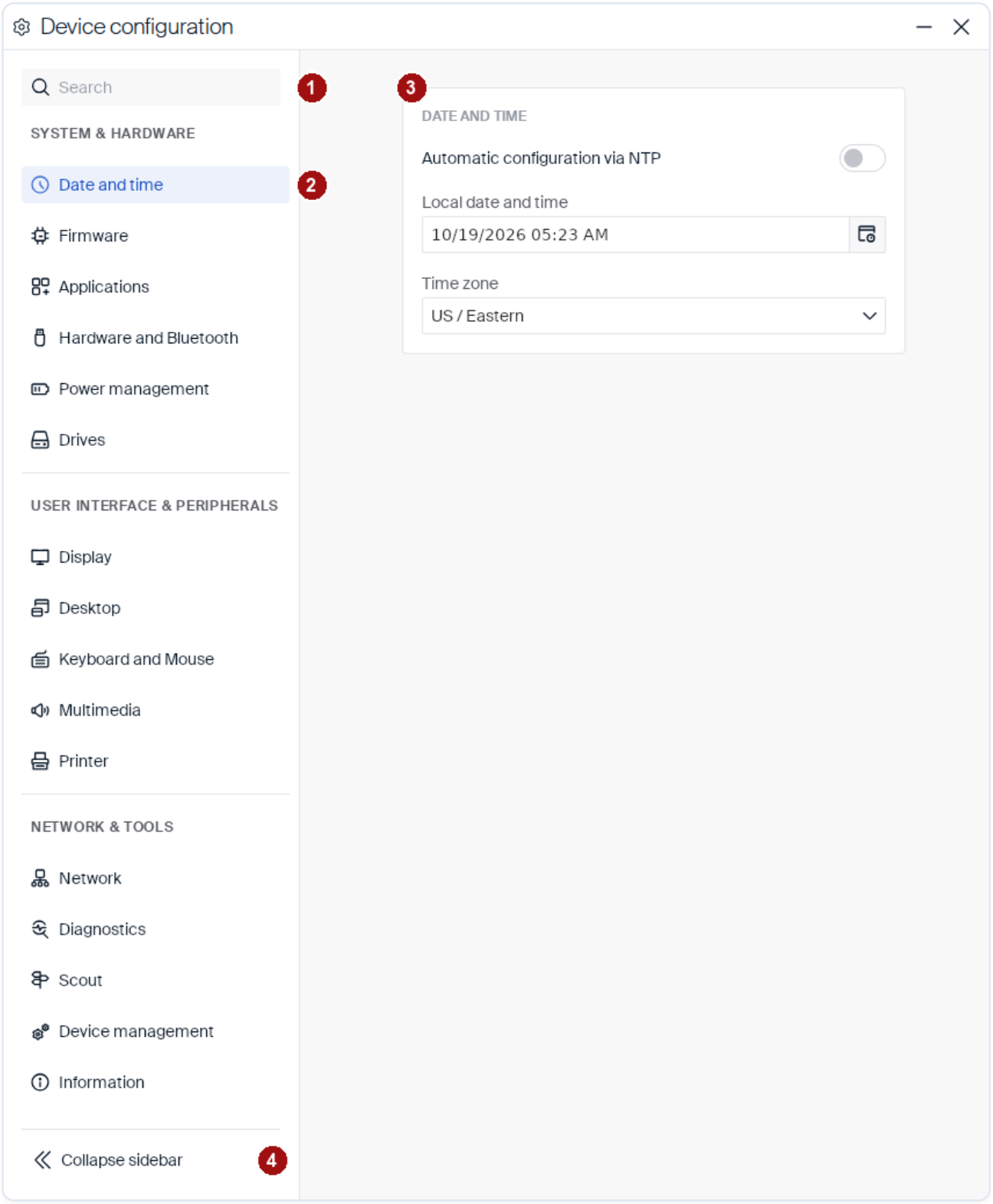
REQUIRES
The option **Show device configuration icon** is enabled (see Desktop).

Open **Device configuration** in one of these ways:

- In the **Start menu**, click the gear icon. **Device configuration** opens on **Date and time**.
- On the system bar, click the time, and then click the gear icon in the **Date and time** popup. **Device configuration** opens on **Date and time**.
- On the system bar, click one of the live information icons, and then click the gear icon in the Quick system access panel. **Device configuration** opens on **Multimedia**.
- On the keyboard, press WIN+ALT+C. **Device configuration** opens on the section you used last.

> **Note:**
>
> If **Show device configuration icon** is disabled, the gear icon in the **Start menu** and WIN+ALT+C do not open **Device configuration**, and clicking the time no longer opens the **Date and time** popup. The **WiFi** and **Wired (LAN)** icons are removed from the system bar, and **Quick system access** shows only the brightness slider and the **Bluetooth**, **Preview** and **Device identifier** tiles.

> The sections are arranged in three groups: **System & Hardware**, **User Interface & Peripherals** and **Network & Tools**.



The numbers in the screenshot mark these parts of **Device configuration**:

	DESCRIPTION
1	**Search** field. Shows only the sections that match your entry.
2	List of sections. Click a section to open it.
3	Settings of the selected section.
4	**Collapse sidebar** reduces the list to icons. Click **Expand** to show the full list again.

Using the sections

- To view all section names or all options of a section, you might have to scroll down.
- Many options are applied directly after you have set them. Some sections require you to click buttons such as **Apply** or **Cancel** before they are closed.
- When you enable an option, further entries might be required.
- If a required field is empty, it is outlined in red and the message **Cannot be empty** is displayed.

For further information about the content of the sections, see Device configuration and Applications.

Applications in the eLux® 7 interface

This article covers how to start, switch between, close, and search for applications from the eLux desktop.

Starting applications

- Double-click an application icon on the desktop to start it. A single click only selects the icon.
- In the application selection in the **Start menu**, click an application to start it.
- Using the keyboard, move to an icon with the arrow keys and press **Enter** (not **Space**) to start it.

Storefront and container icons always start on a single click, regardless of the desktop double-click setting.

To switch back to single-click (as in earlier eLux versions), configure the following:

File	`/setup/terminal.ini`
Section	`Desktop`
Key	`DoubleClickAppStart`
Values	`true` (default) / `false`

Set this through **Scout** → **Advanced Configuration** → **File Entries**, then send a configuration update and restart the device. With `false`, a single click starts a desktop application again, as in earlier versions.

> **Note:**
>
> On touch devices, set `DoubleClickAppStart` to `false`. Double-tap is unreliable on a touchscreen, and with the default setting, a single tap does not start anything.

Disconnecting applications

- On the system bar, right-click the open application and then click **Close**.

> **Note:**
>
> To close a session completely, the user must log off. In addition, the administrator can define a timeout on the server to close any inactive sessions.

Searching for applications

- Enter the search term in the search field of the application search in the Start menu.

Only the relevant applications are displayed in the application selection.

- To show all applications again, delete the text in the search field.

Switching between sessions/applications and the eLux Desktop interface

- Press the key combination CTRL+ALT+D.

Showing the desktop and minimizing application windows

- On the system bar, click the **Show desktop** icon .

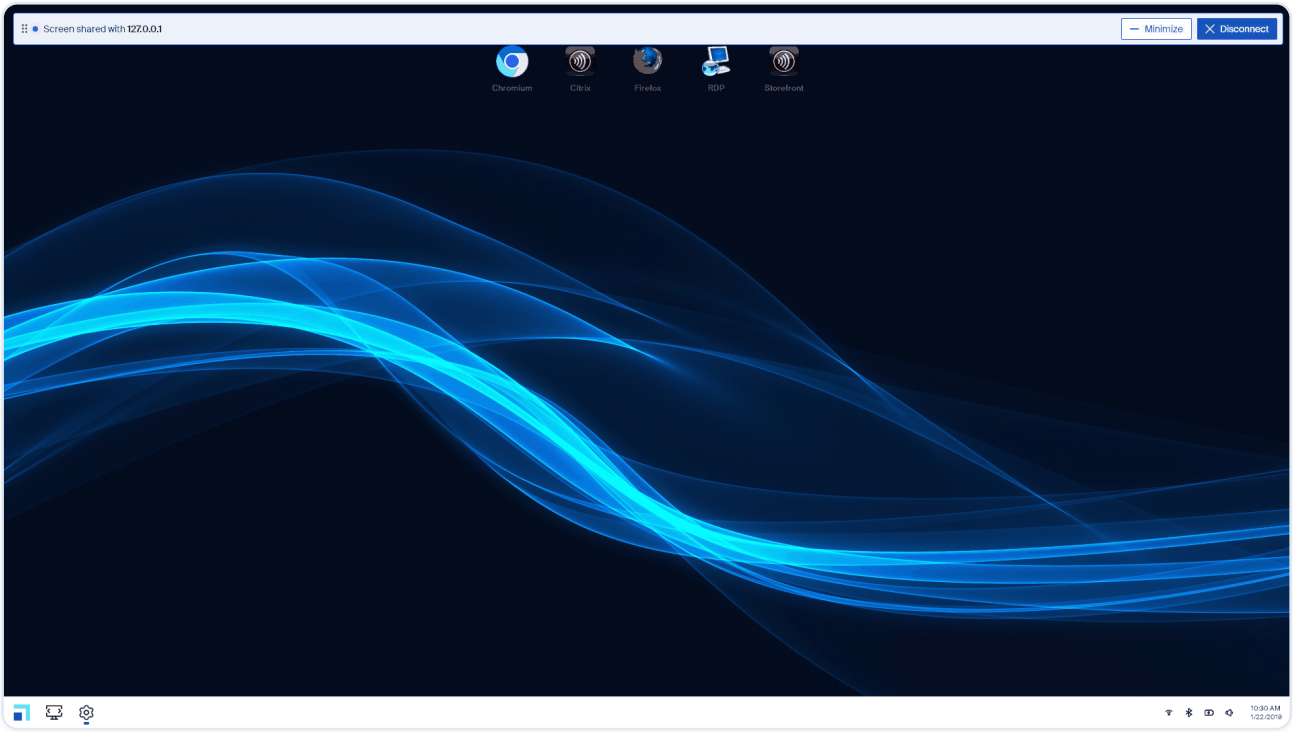
For further information, see Applications.

Mirror session bar

Mirroring lets a Scout administrator view your device's screen remotely, typically to help with troubleshooting.

When your device is mirrored, a bar appears to let you know your screen is being shared.

You can move it out of the way or collapse it to a small pill so it doesn't cover the icons, menus, and controls you need.



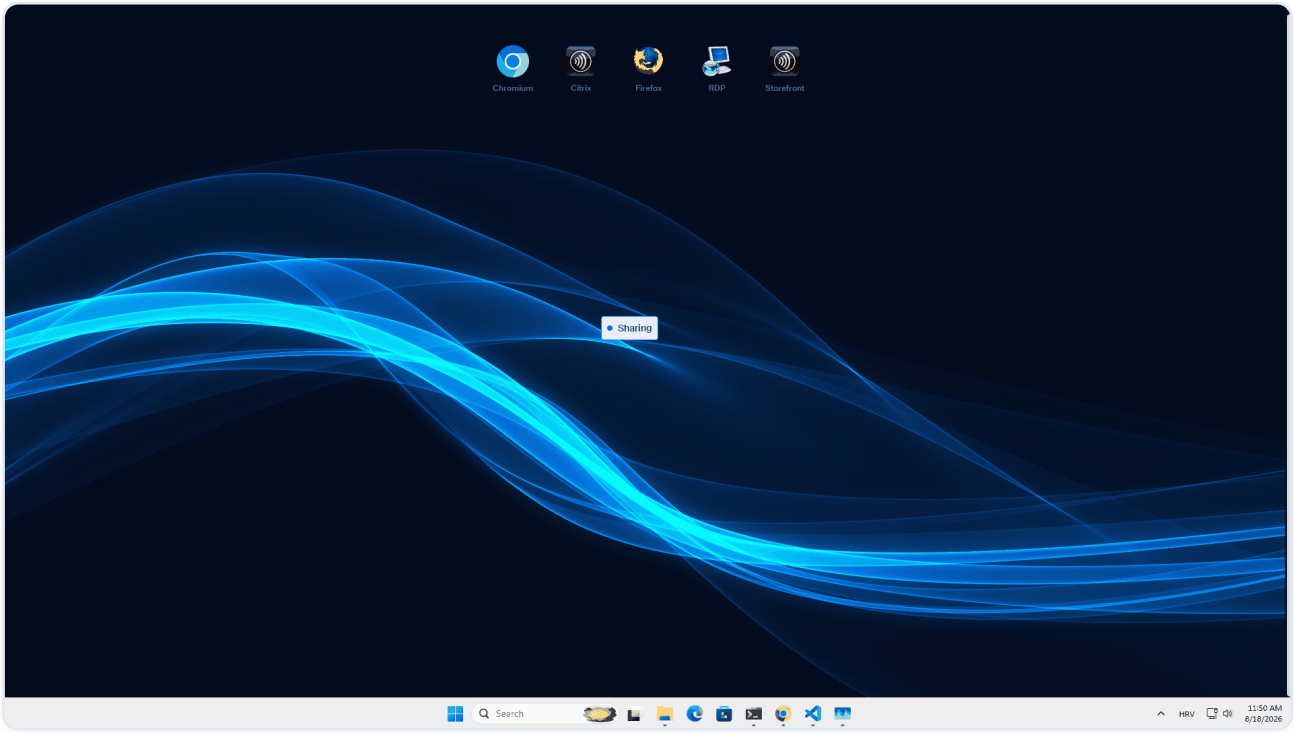
The bar includes:

ELEMENT	DESCRIPTION
Pulsing blue dot	Mirroring is live.
"Screen shared with `<IP address>`"	The IP address of the administrator's session.
Six-dot handle (left)	Drag to collapse the bar into a movable pill.
Minimize (right)	Used to collapse the bar.

ELEMENT	DESCRIPTION
Disconnect	Ends the mirror session.

Moving it out of the way

- **Minimize**: click **Minimize** — the bar collapses to a pill labeled **Sharing** in the top-right corner.
- **Drag it away**: grab the six-dot handle on the left and drag — it collapses into the pill and follows your cursor to wherever you drop it.
- **Move the pill**: drag it from any point once it's collapsed. Hovering over it fades in the six-dot handle so you can see it's movable.
- **Restore the full bar**: click the pill (without dragging).



Good to know

- The bar and pill appear on the **primary monitor** only, and stay on top of everything else, including a full-screen Citrix® session.
- The expanded bar stays docked at the top; only the collapsed pill can be moved freely, and it's always kept fully on screen.
- Position and collapse state are **not remembered** — every new mirror session starts with the full bar docked at the top.
- Ending the mirror session removes the bar or pill.

- An administrator can mirror the device only if **Mirroring enabled** is turned on for it in UniconOS Management Board, in the **Mirroring** section of the device configuration. To start a mirror session from UniconOS Management Board, see Mirroring a device.

Device information overlay

The **device information overlay** is a small, semi-transparent panel in the top-right corner of the desktop wallpaper.

It shows identifying details about the device — name, network address, logged-in user, and the managing Scout server.

This helps helpdesk staff identify a device during a remote support call without asking the user to look anything up.

The panel is **off by default**. An administrator turns it on per device or OU.

For a similar tile shown before sign-in, on the logon screen itself, see Logon screen information.

Turning it on

With UniconOS Management 2609 (Scout 15.2609) and later, configure the overlay centrally in UniconOS Management Board: **Devices** > **Device management** > **Device configuration** > **Desktop** > **Show desktop info overlay**. UniconOS Management Board also lets you choose the fields and theme, and add custom entries, with no file entries needed. For details, see Device configuration / Desktop in the UniconOS Management Board guide.

For earlier Scout versions, set this in **Scout** → **Advanced Configuration** → **File Entries**:

File	`/setup/terminal.ini`
Section	`Layout`
Key	`EnableDeviceInfoOverlay`
Values	`false` (default) / `true`

An advanced file entry for one of the keys on this page overrides the UniconOS Management Board Desktop setting for that key. If you switch a device to the UniconOS Management Board route, remove any matching file entries so UniconOS Management Board's values are not silently overridden.

Changes take effect after a restart — overlay settings are read when the desktop starts.

What it shows

By default, the panel lists all nine fields, grouped into sections:

SECTION	FIELDS
System	Device name, Domain, Operating system, Boot time
Network	IPv4, MAC
User	Logged in user
Scout server	Scout server, Manager address

A row is left out when the device has no value for it — for example, **Logged in user** only appears once someone has logged in.

If nothing can be shown, the panel doesn't appear.

To show only some fields, list them as a comma-separated set:

File	`/setup/terminal.ini`
Section	`Layout`
Key	`DeviceInfoOverlayFields`
Values	*empty* (default, shows all nine) or a comma-separated list

Field names: `username, hostname, domain, ip, mac, os, scoutManagerName, scoutManagerAddress, bootTime`. Names are case-sensitive.

For example, `hostname,ip,username` shows only the device name, IPv4 address, and logged-in user.

A section disappears when none of its fields are selected; an unrecognized name is simply ignored.

Adding custom fields

Add your own rows — an asset tag, a room number, a support number — in an extra **Additional** section:

File	`/setup/terminal.ini`
Section	`Layout`
Key	`DeviceInfoOverlayCustomFields`
Values	`Label=Value;Label2=Value2`

Use `=` between a label and its value, and `;` between rows:

```
Location=Building 7, Floor 2;Asset tag=UC-004182;Support=x4400
```

Only the first `=` on each row is the separator, so values can contain `=` too.

A row with an empty label or value is skipped, not broken.

Custom fields always appear, even if `DeviceInfoOverlayFields` narrows the standard ones.

Good to know

- The panel sits in the **top-right corner** and appears **after login**; its position is fixed.
- It's **click-through** — clicks pass to whatever is underneath, so it never gets in the way.
- Desktop icons flow around it: they line up from the left edge, leaving the area under the panel free.
- On an extended desktop, the panel appears on **each monitor**.
- **Boot time** uses the date and time format of the desktop language.
- **IPv4 and MAC** update live if the network changes — no restart needed for those two fields.
- Switch between light and dark themes with `DeviceInfoOverlayTheme` (same `Layout` section) — `light` (default) is pale with dark text, `dark` is the reverse.

Logon screen information

Logon screen information is a tile shown on the UniconOS logon screen, before anyone signs in.

It lists the device name, IP address, organizational unit (OU), and firmware version, plus up to ten entries of your own, such as a helpdesk phone number or an asset tag. Users and helpdesk staff can identify a device without signing in first.

The tile is **off by default**. An administrator turns it on centrally in UniconOS Management Board.

> **Note:**
>
> This is a separate feature from the device information overlay, which appears on the desktop after sign-in.

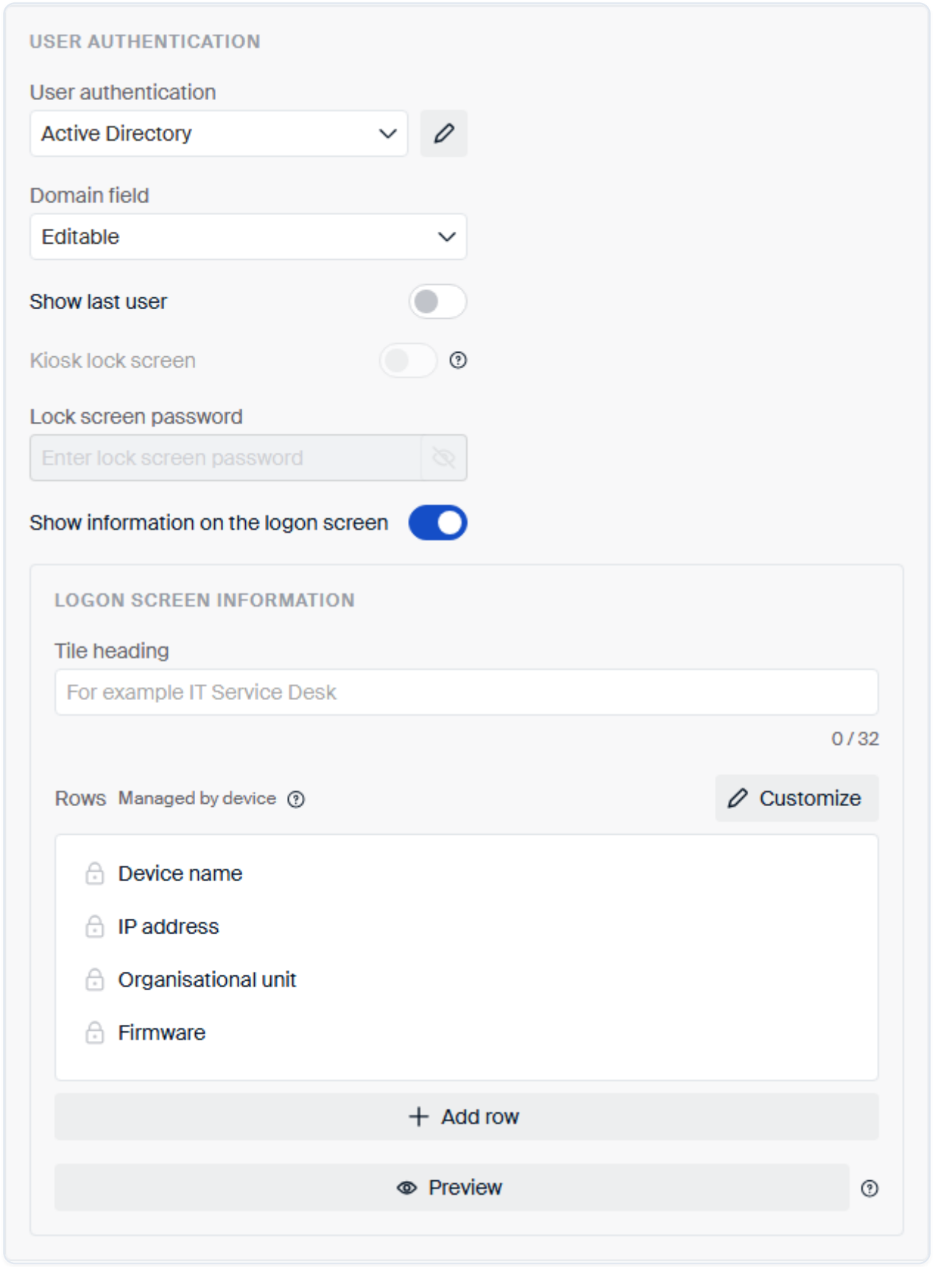
Turning it on

Open the device configuration of the base configuration, an OU, or a device:

- **Base configuration**: click **Devices** > **Device management** > **Device configuration**.
- **OU or device**: open its context menu and click **Device configuration**. For details, see Device configuration.

Then turn on the tile:

1. For an OU or a device, clear **Use parent device configuration**. **Edit** stays disabled until you do.
2. Click **Edit**.
3. In the list of sections, click **User authentication**.
4. Turn on **Show information on the logon screen**. The switch is below **Lock screen password**.
5. Set up the tile as described in Configuring the tile.
6. Click **Apply**. On a device, click **Apply settings**.


USER AUTHENTICATION
User authentication
Active Directory
Domain field
Editable
Show last user
Kiosk lock screen
Lock screen password
Enter lock screen password
Show information on the logon screen
LOGON SCREEN INFORMATION
Tile heading
For example IT Service Desk
0 / 32
Rows
Managed by device
Customize
Device name
IP address
Organisational unit
Firmware
Add row
Preview

Note:

> Clearing **Use parent device configuration** takes effect at once. It stays in effect even if you discard your other changes. The OU or device gets its own copy of the **whole** configuration, so later changes on the parent no longer reach it, for any setting. Selecting the checkbox again deletes that copy after you confirm.

> Configuration labels can carry these settings too. In a label, search for **User authentication** to add the whole section, or for **Logon screen information** to add only the tile. A label overrides only the settings it carries. If a label carries the tile without the sign-in type, UniconOS Management Board can't tell whether the tile will appear. It shows a note that lists the sign-in types that show the tile.

The legacy Scout Console has no controls for this feature. Console users can set the same `/setup/terminal.ini` keys as advanced file entries.

> **Note:**

> Administrators need the **Security** right to see the User authentication section, and **Edit user authentication** to change its values.

Configuring the tile

The tile has a heading and a list of rows. You set both in the **Logon screen information** section:

- **Tile heading**: optional, up to 32 characters. A counter under the field shows how many you have used. A line break in a pasted heading becomes a space. If you leave it empty, the device shows **Device information** in its display language.
- **Rows**: device fields first, then your custom entries, below a divider labeled **Custom entries - always rendered below**. Drag a row by its handle to reorder it. The list order is the display order. You can't drag a row across the divider.

UniconOS Management Board offers these device fields. The table also shows the label each field has on the device tile:

UNICONOS MANAGEMENT BOARD LABEL	LABEL ON THE DEVICE	SHOWS
Device name	**Device**	The device's host name

UNICONOS MANAGEMENT BOARD LABEL	LABEL ON THE DEVICE	SHOWS
IP address	**IP address**	The device's IPv4 address on its own network
Organisational unit	**Organisational unit**	The OU name and number, for example `Sales (42)`
Firmware	**Firmware**	The operating system name and version
Scout server	**Scout server**	The name of the Scout server that manages the device

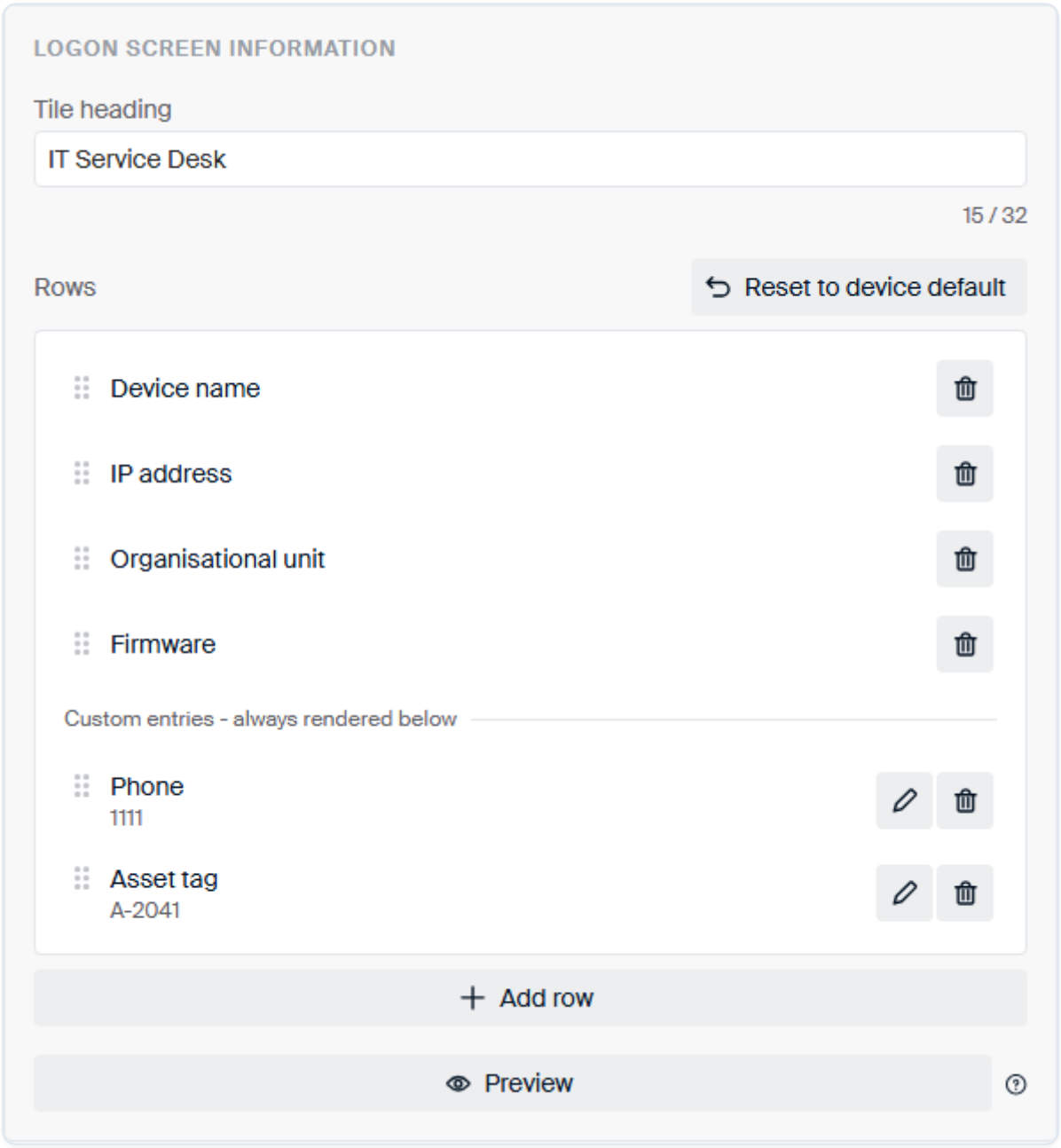
If the OU name can't be looked up, the **Organisational unit** row shows only the OU number.

By default, the rows are **Managed by device**. The tile shows the device name, IP address, OU, and firmware, and the rows are locked. The device chooses these rows, so a field that a later firmware version adds appears on its own.

Click **Customize** to unlock the rows and build your own list. Click **Reset to device default** to go back to **Managed by device**. This also removes your custom entries.

Click **Add row** to add a device field that isn't in the list yet, or a **Custom entry ...**. A counter shows how many of the 10 custom entries are left. Adding a device field also unlocks the rows.

Every row has a drag handle and a **Remove** button. Custom entries also have an **Edit** button. To reorder with the keyboard, press Space on the handle, move with the arrow keys, and press Space again to drop the row.



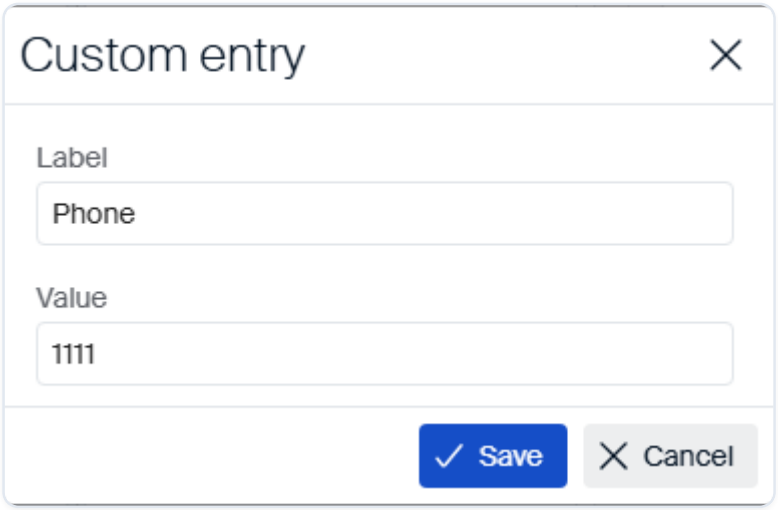
If you remove every device field, UniconOS Management Board shows "No device fields. Only your custom entries are shown." The tile then shows only your custom entries. The default fields don't come back.

In the **Custom entry** dialog, enter a **Label** and a **Value**:

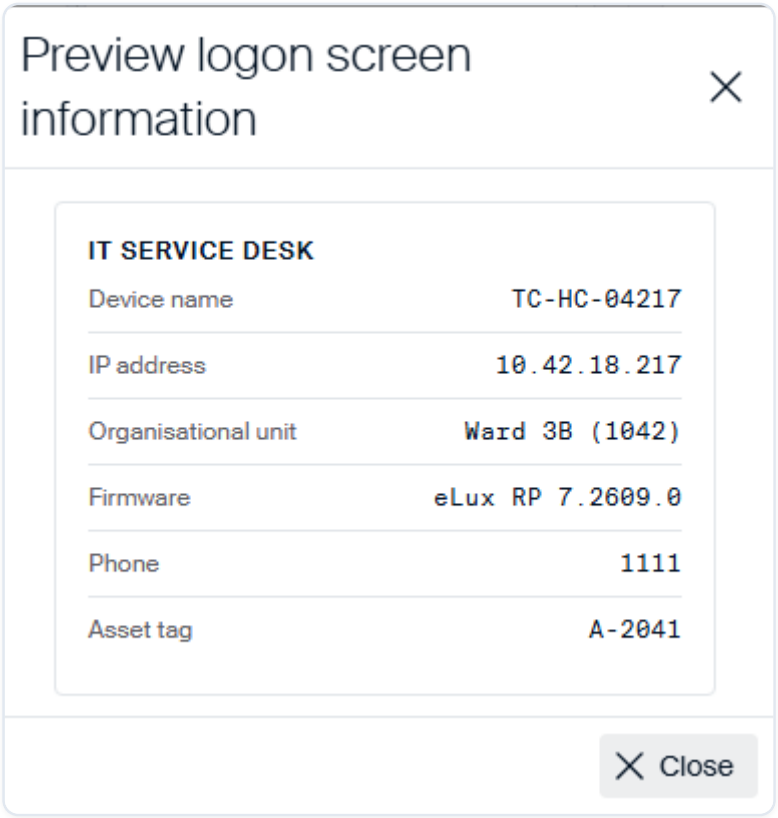
- **Label**: required, up to 32 characters, and no `=`.
- **Value**: required, up to 128 characters. It can contain `;` and `=`.

Custom entries are static text. There is no placeholder syntax, and the value is shown exactly as you typed it.

If an entry breaks one of these rules, the dialog shows an error such as "The label is longer than 32 characters." and doesn't save the entry. It doesn't shorten the text for you. The UniconOS Management Board API rejects such entries with HTTP 400.



Click **Preview** to see the tile with sample values, not the device's real data. The preview labels the host name row **Device name**, but the device shows **Device**. About six rows fit on a 1366 x 768 screen.



If **User authentication** is set to **None**, **Evidian**, or **Citrix®**, UniconOS Management Board shows a warning such as "None sign-in does not draw this tile. Anything set here is saved but has no effect on the device." It doesn't block saving.

On the device

- The tile appears on the logon screen, in the branding area on the left, under the product logo.
- It appears only for these sign-in types: **Active Directory**, **Active Directory + Smartcard**, **OpenID Connect**, **Entra ID**, **Imprivata**, and **SAML**.
- The logon screen must be at least 1200 pixels wide after display scaling. Below that, the logon screen has no branding area to hold the tile. For example, a 1920 x 1080 screen at 200% scaling counts as 960 pixels, so no tile is shown.
- A device field with no value is left out. For example, a device that isn't in an OU has no **Organisational unit** row. If no rows are left, no tile appears — a heading on its own doesn't show anything.
- Built-in labels and the default heading are translated; your own heading and custom entries aren't, and appear exactly as entered. The heading — default or your own — is always shown in capital letters; custom entries are not.
- Long values wrap.
- Devices get changes with their next configuration update from Scout.
- The device needs UniconOS 2609 or later. Earlier versions don't show a tile.
- For devices that connect through the Scout Cloud Gateway, the **IP address** row shows the device's own network address, which can differ from the address Scout shows for the device.

Advanced file entries

For Console users, set these in `/setup/terminal.ini`, section `Layout`:

KEY	VALUE
`EnableLoginInfo`	`true` or `false` (default)
`LoginInfoFields`	Comma-separated device fields, in display order. Default: `hostname,ip,ou,os`.
`LoginInfoTitle`	The tile heading. If empty, the device shows **Device information**.
`LoginInfoCustomField1` through `LoginInfoCustomField10`	`Label=Value`, split at the first `=`.

In `LoginInfoFields`, use these case-sensitive names: `hostname`, `ip`, `ou`, `os`, and `scoutServer`. Use `-` for no device fields.

Configure the tile in one place only. The UniconOS Management Board panel and advanced file entries write the same keys. Scout sends advanced file entries after the panel's values, and the device keeps the last one, so an entry overrides the panel wherever file, section, and key match. Scout also sends all ten `LoginInfoCustomFieldN` keys every time, so an entry replaces whatever the panel put in that slot.

UniconOS Management Board's length and count limits apply to the panel and the UniconOS Management Board API. Scout doesn't check advanced file entries for them.

An entry with no `=` shows as a full-width line without a label, for example "Please quote the device name when raising a ticket". The panel doesn't offer label-less entries.

See Advanced file entries for how to set a file entry.

Kiosk lock screen

Kiosk lock screen locks a shared kiosk device — one with no user authentication — behind a single administrator-set password, instead of leaving an idle session open to anyone who walks up to it.

> **Note:**
>
> Configure this feature in UniconOS Management Board: in the **User authentication** section of the device configuration, turn on **Kiosk lock screen** and enter the **Lock screen password**. See Kiosk lock screen in the **UniconOS Management Board** guide.

On the device

The device must already be running in kiosk mode (UC Self Service or UC Browser) with no user authentication. The feature doesn't apply to a plain no-authentication desktop, Boot-to-VDI, or Boot-to-App.

With the kiosk lock screen turned on, the device locks in these cases:

- The device is idle until the screen saver starts, see **Enable screen saver after** in Power management.
- A user presses the key combination for locking the screen, Ctrl+Alt+End by default.
- A user clicks **Lock** in the **Start menu**. Kiosk mode doesn't hide the system bar, so the **Start menu** is available.
- The device goes into sleep mode. It wakes up on the lock screen.

An application that keeps the screen awake, for example a web page that plays a video, prevents the lock for inactivity. The other cases still lock the device.

The lock screen shows one instruction line and a single input field:

Instruction line	"Please enter the password to unlock."
Field	**Password** only — no username or domain field. The password belongs to the device, not to a person.

Entering the correct password returns to the session that was already running — it doesn't start a new one. Entering an incorrect password clears the field and shows "The password is incorrect." The device stays locked, and you can try again immediately — there's no lockout after repeated wrong attempts.

If the device has an Imprivata bootstrap URL configured, Imprivata's own lock screen takes over instead, and this feature has no effect.

Desktop and system bar sizes

Desktop and system bar sizes are three settings that scale the desktop icons, the text under them, and the system bar. Use them to adapt a device to accessibility needs or to a particular screen, without turning on system-wide display scaling.

The settings need UniconOS 2609 or later. Scout and UniconOS Management Board have no dedicated setting for them, but both let you set them as advanced file entries. For how to define an entry, see Advanced file entries. Devices get a change with their next configuration update from Scout.

All three keys are in the `Desktop` section of `/setup/terminal.ini`. Each value is a whole number of pixels:

KEY	DEFAULT	RANGE	APPLIES TO
`DesktopIconSize`	64	32–128	Custom grid layout only
`DesktopFontSize`	14	10–24	Custom grid layout only

KEY	DEFAULT	RANGE	APPLIES TO
`SystembarHeight`	48	40–128	Both desktop layouts

Desktop icons and labels

`DesktopIconSize` sets the size of the application icons on the desktop. `DesktopFontSize` sets the size of the text under them. The icon tile and the label box scale with these values, so longer application names wrap onto two lines instead of being cut off.

> **Note:**

> The custom grid layout is on only when `UseCustomGridLayout=true` in the same section. It is off by default. On the standard desktop layout, `DesktopIconSize` and `DesktopFontSize` do nothing, whatever you set. If a new icon size changes nothing, check `UseCustomGridLayout` first.

System bar height

`SystembarHeight` sets the height of the system bar. Unlike the icon and font keys, it also works on the standard desktop layout.

The desktop, the tray, and the quick-access panels move with the bar, so nothing is hidden behind it. The buttons inside the bar keep their normal size. A taller bar adds space around them instead of enlarging them.

Values outside the range

The device never rejects a value and never shows an error. It uses the nearest allowed value instead:

VALUE YOU ENTER	RESULT
Below the range, `0`, or negative	Raised to the minimum. `DesktopFontSize=2` shows as `10`.
Above the range	Lowered to the maximum. `DesktopIconSize=500` shows as `128`.
Decimal number	Cut to a whole number, not rounded. `SystembarHeight=47.6` gives a 47 px bar.

VALUE YOU ENTER	RESULT
Key missing or empty	The default is used.
Text instead of a number	Read as `0`, so raised to the minimum. `SystembarHeight=48px` gives 40 px, not 48 px.

> **Note:**
>
> The device stores the value you entered, not the value it uses. If you set `SystembarHeight=999` and read the value back, it still reports `999`, although the bar is 128 px high. Judge the result by what you see on the device.

On-Screen Keyboard

eLux can show an on-screen keyboard so a device without a physical keyboard can still receive input. It's available on the logon screen, the lock screen, and the desktop.

Before you start

The device image must contain the **onboard** on-screen keyboard package, which the standard images don't contain. Without it, the keyboard never appears, no matter how it's configured, and the device configuration doesn't show **Accessibility**.

On UniconOS 2611, onboard also needs these feature packages, which the standard images leave out: **gir_libs** and **iso_codes** in the **desktop_environment** package, and **libhunspell** in the **systemlibs** package. You add packages and feature packages to an image in **Software** in UniconOS Management Board, see Changing an image.

Configuring it from UniconOS Management Board

On UniconOS 2609 and later, configure AutoStart, Theme and Layout centrally from UniconOS Management Board instead of with file entries — see Keyboard and Mouse. On earlier eLux versions, use the file-entry method below.

An administrator can also lock the device user out of the settings below, see User rights. When **Enable virtual keyboard** is locked, the **Accessibility** toggle described below disappears. A device user can still close an open keyboard with its close button, but then has no control to reopen it until the device restarts. To prevent this, also remove the close button, see Preventing users from closing the keyboard.

Turning it on and off

A user can show or hide the keyboard at any time from **Quick System Access → Accessibility**, unless an administrator has locked this. The toggle stays in sync with the keyboard's actual state — for example, if the user closes the keyboard window directly, the toggle switches off too.

By default, this is a per-session choice only: the keyboard's state resets the next time the device reboots.

Making it start automatically

To have the keyboard come up by itself after every reboot — at the login prompt, on the desktop, and on the lock screen — set the following in **Scout → Advanced Configuration → File Entries**:

FILE	SECTION	KEY	VALUE
`/setup/terminal.ini`	`VirtualKeyboard`	`AutoStart`	`true` (starts automatically on every boot) / `false` or omitted (today's behavior — off by default)

This is a policy the administrator sets centrally; there's no per-device or per-user override on the device itself, and the on-screen toggle in Accessibility does not rewrite it. It's off by default and available from eLux 7.2607.

Push it to a device or an OU, then reboot for it to take effect. Once set, hiding the keyboard for the current session (via Accessibility) doesn't change this setting — the next reboot re-applies it, and the keyboard comes back if `AutoStart=true`.

Changing the layout and theme

Configure the keyboard's appearance the same way, via **Advanced Configuration → File Entries**:

FILE	SECTION	KEY	VALUES
`/setup/terminal.ini`	`VirtualKeyboard`	`Theme`	`elux-dark` / `elux-light`

FILE	SECTION	KEY	VALUES
`/setup/terminal.ini`	`VirtualKeyboard`	`Layout`	`Compact-Elux` / `Full Keyboard-Elux`

On UniconOS 2609 and later, a device user can also change the theme and the layout directly on the device, in the device configuration under **Accessibility** > **Onscreen keyboard settings** while the keyboard is shown, unless an administrator locks the matching user right.

Preventing users from closing the keyboard

On a touch-only device without a physical keyboard, the on-screen keyboard is the only way to type. So that users can't close it with a stray tap, you can remove its close button (X). The keyboard then has no close button on the logon screen, the desktop and the lock screen. The place of the close button stays empty, and the other keys don't move. From UniconOS 2611, the keyboard also has no key that quits it, whether or not the close button is shown.

Removing the close button doesn't remove the switches that show and hide the keyboard. Unless an administrator has locked them, users can still hide the keyboard on purpose. They can use **Show on screen keyboard** in the device configuration, or **Quick System Access** > **Accessibility**, also on the logon and lock screens. To have the keyboard come up by itself after every restart, also make it start automatically, see Making it start automatically.

The close button is shown by default. Removing it needs UniconOS 2611 or later, with the **onboard** package of the same release in the image. With an older **onboard** package, **Hide keyboard close button** is shown but has no effect. Earlier UniconOS versions ignore the setting.

ON THE DEVICE

In the device configuration, open **Accessibility** and turn on **Hide keyboard close button**. You can do this whether or not the keyboard is shown. The close button disappears at once, also from a keyboard that is already open. The setting stays after a logoff and a restart. To bring the close button back, turn the switch off.

If an administrator has locked the setting, the device configuration doesn't show the switch. On a device that Scout manages, set the value in Scout: from UniconOS Management 2611, the next configuration update from Scout normally replaces a change made on the device.

IN UNICONOS MANAGEMENT BOARD

With UniconOS Management 2611 (Scout 15.2611) and later, turn on **Prevent closing the virtual keyboard** in Scout Board, in the device configuration under **Keyboard and Mouse** > **Virtual**

keyboard. You can set it in the base configuration, for an OU, for a device and in configuration labels. To stop users from changing it on the device, also lock the user right **Prevent closing the virtual keyboard**. For details, see Keyboard and Mouse.

WITH AN ADVANCED FILE ENTRY

Scout 2609 and earlier have no setting for it. Use an advanced file entry instead. In UniconOS Management Board, you find advanced file entries in the **Advanced file entries** section of the advanced device configuration.

To remove the close button, push this key to a device or an OU:

FILE	SECTION	KEY	VALUE
`/setup/terminal.ini`	`VirtualKeyboard`	`PreventClose`	`true` (no close button) / `false` or omitted (close button shown, default)

The device applies the value the next time it starts. An advanced file entry is not a lock. Users can still change **Hide keyboard close button** on the device. The value of the entry comes back only when the device next gets its configuration from Scout. If you later manage the setting in UniconOS Management Board, delete the file entry, because a leftover entry can override the UniconOS Management Board switch.

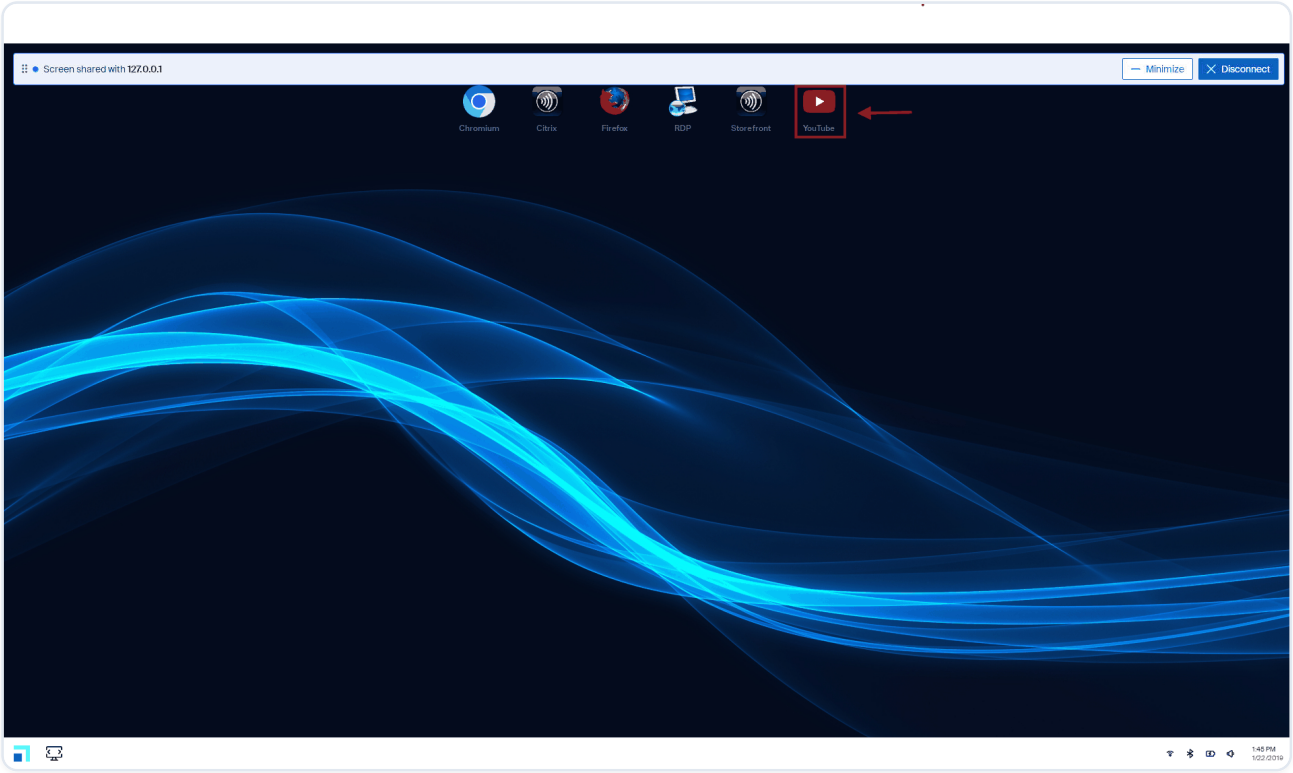
Progressive Web App (PWA) support

Progressive Web Apps (PWAs) are websites you install as apps from your browser.

Once installed, a PWA appears on the system bar and desktop with its own icon — you can start it, switch to it, and close it like any other app.

No configuration is needed; this works out of the box.

FIGURE: YOUTUBE



Installing a PWA

In your browser:

1. Open the site you want, then install it as an app (⋮ → **Cast, save and share** → **Install page as app**, or the install icon in the address bar).
2. The app appears on the system bar and desktop within a few seconds, with its own icon.
3. Click the icon to start it — click it again while it's running to bring its window to the front.

Removing a PWA

Uninstall it from the **browser**, not from eLux®.

Open the browser's apps page, right-click the app, and remove it — the icon disappears from the system bar and desktop within a few seconds:

BROWSER	APPS PAGE
Chrome / Chromium	`chrome://apps` → right-click the app → **Remove from Chrome**
Microsoft Edge	`edge://apps` → **…** → **Uninstall**
Island	The browser's apps page (same pattern as Chrome)

Apps your administrator installs by policy can't be removed this way — they're reinstalled automatically.

Turning it off (administrators)

PWA support is **on by default**. Set this in **Scout → Advanced Configuration → File Entries** to turn it off for a device or OU:

File	`/setup/terminal.ini`
Section	`Global`
Key	`EnablePwaApps`
Values	`true` (default) / `false`

The setting applies after a restart.

> **Important:**

> Turning this off is **destructive**: existing PWA launchers and icons are deleted from the device, and turning it back on does not restore them.

> To use a previously installed PWA again, uninstall it from the browser's apps page (for example `chrome://apps`) and reinstall it.

Good to know

- Works with Chromium-based browsers (Google Chrome, Microsoft Edge, Chromium, Island). Firefox doesn't create app entries.
- To keep installed apps after a restart, include the browser in **Persistent home applications** (`PersistentHomeApplications`, or set it to `ALL`).

Device configuration

> **Important:**

> If the device is managed by Scout, configuration is normally done centrally in UniconOS Management Board. With inheritance enabled, local configuration changes on the device are overwritten as soon as the device connects to Scout. For further information, see Device configuration in the **UniconOS Management Board** guide.

> The local device configuration is located in **Device configuration**. For further information on opening and using Device configuration, see Device configuration.

Applications

Multimedia

Date and time

Desktop

Device management

Diagnostics

Display

Drives

Firmware

Hardware and Bluetooth

Information

Keyboard and Mouse

Network

Power management

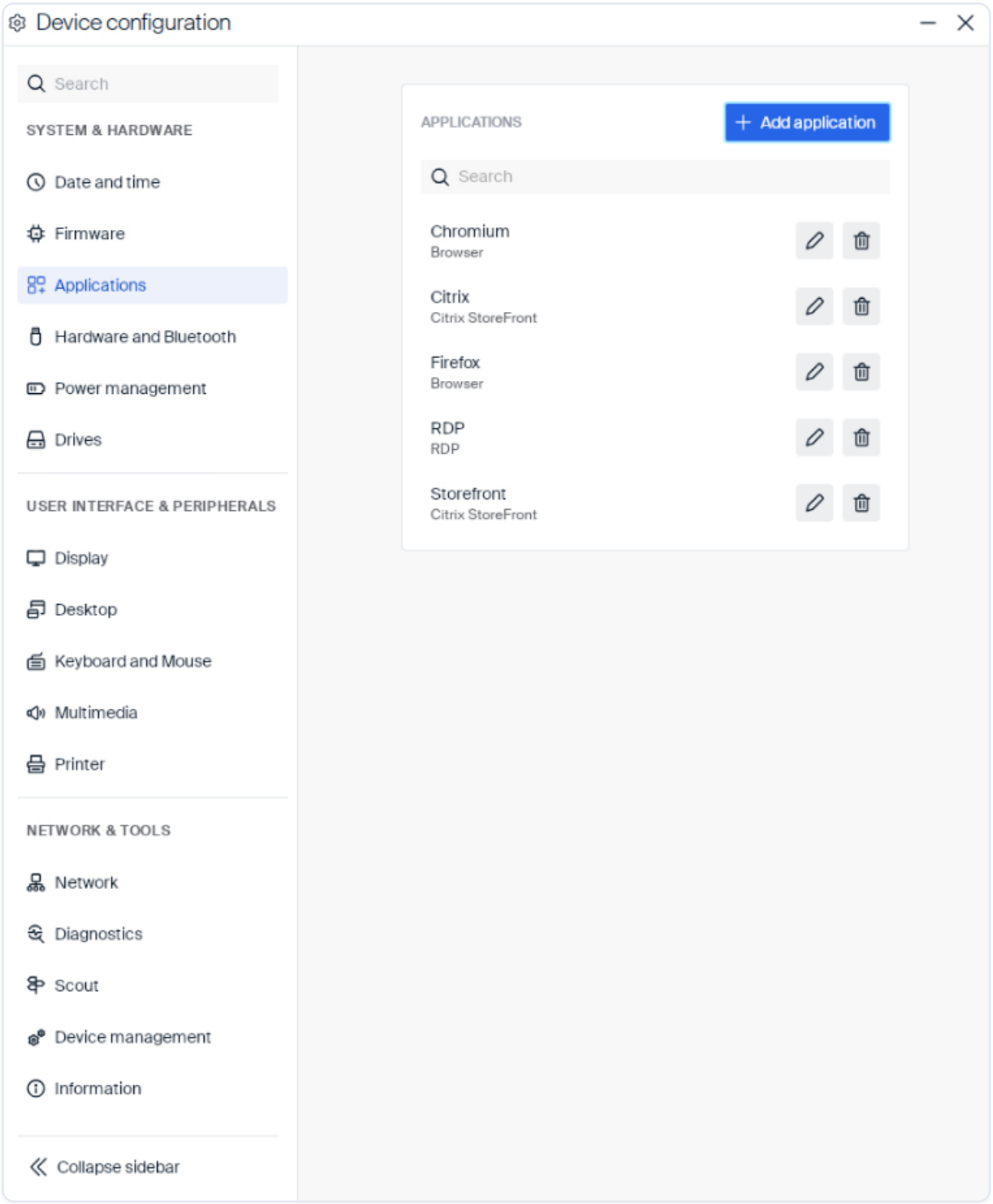
Printer

Scout

Applications

Applications is where you add, edit, and remove the device's programs and remote connections — browsers, Citrix® StoreFront™ resources, RDP and Omnissa Horizon sessions, and local applications.

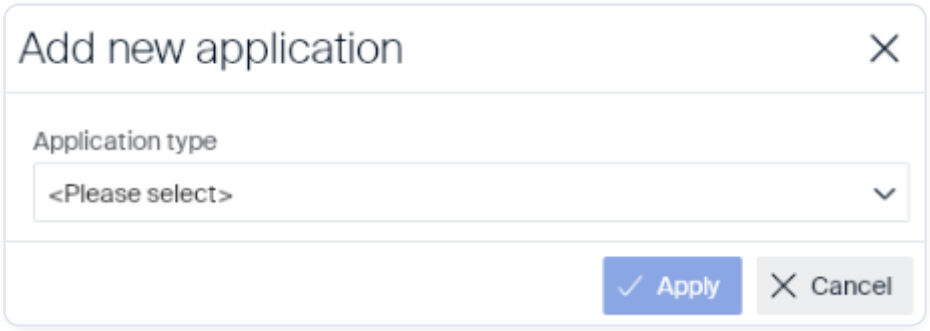
The corresponding software must be installed on the device (for example Citrix Workspace™ app, a web browser, or Omnissa Horizon Client).



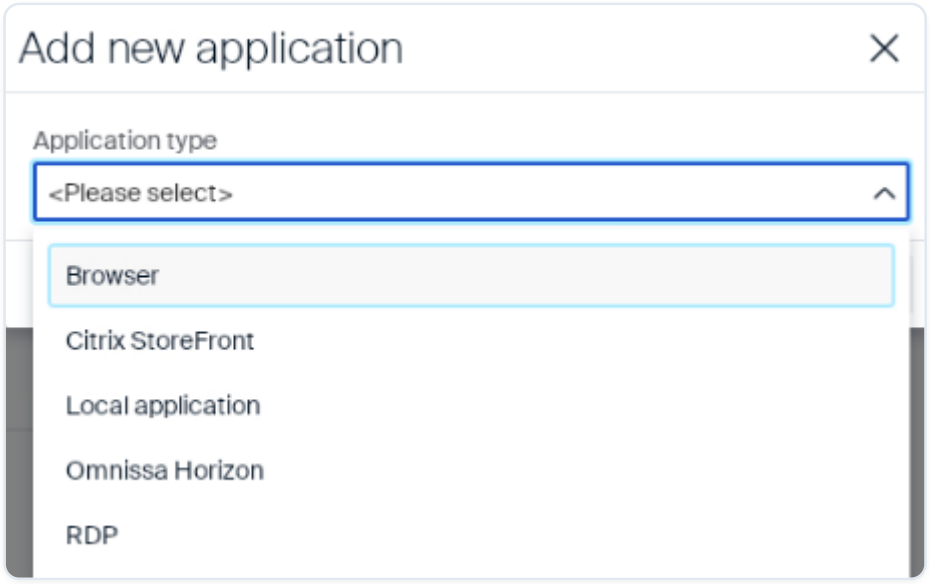
Each row shows the application's name and type. Use the search field to filter the list, the pencil icon to edit an application, and the trash icon to remove it.

Adding an Application

1. Click **Add application**.



2. Open the **Application type** dropdown and select a type: `Browser`, `Citrix StoreFront`, `Local application`, `Omnissa Horizon`, or `RDP`.



3. Fill in the fields for the selected type (see the sections below), then click **Apply**.

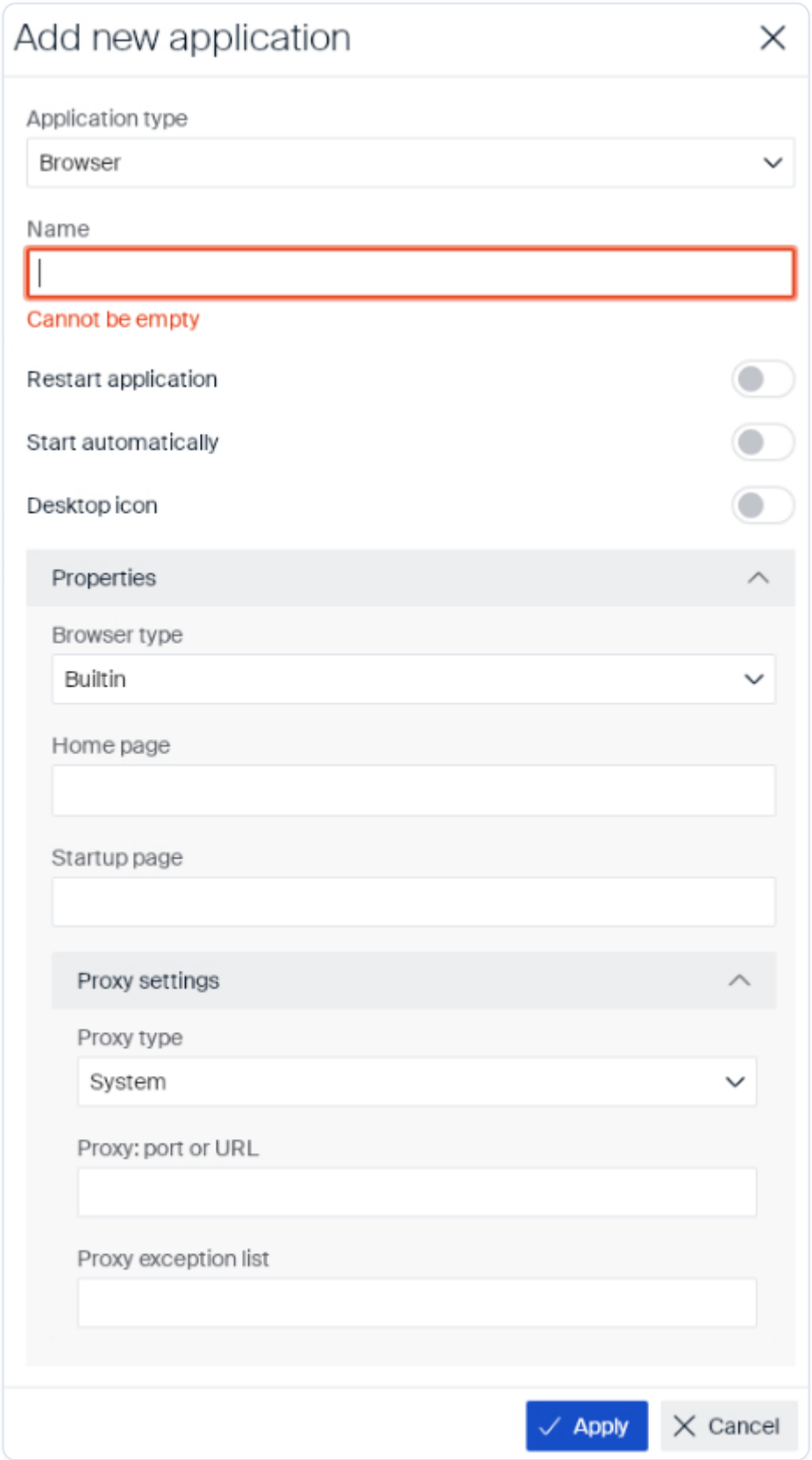
Common Fields

Regardless of type, every application you add or edit includes these fields:

FIELD	TYPE	DESCRIPTION
Name	Text field	Name of the application. Required — cannot be empty.
Restart application	Toggle	Restarts the application immediately if it closes unexpectedly or is closed by the user. Not available for Omnissa Horizon.

FIELD	TYPE	DESCRIPTION
Start automatically	Toggle	Starts the application automatically after eLux® starts.
after (seconds)	Number field	Delay before the automatic start, in seconds. Only shown when **Start automatically** is on.
Desktop icon	Toggle	Adds a shortcut icon for the application on the desktop.

Browser



Besides the common fields, when you add a **Browser** application, you're presented with the following fields, which control which browser engine is used and what page it opens to:

FIELD	TYPE	DESCRIPTION
Browser type	Dropdown	`Builtin` (built-in browser), `Chromium`, `Google Chrome`, or `Firefox`.
Home page	Text field	Web page that opens when the user clicks **Home**.
Startup page	Text field	Web page that opens when the browser starts.

PROXY SETTINGS

Proxy settings determine whether the browser's traffic goes through a proxy server, and how it's configured:

FIELD	TYPE	DESCRIPTION
Proxy type	Dropdown	`Manual (Proxy:Port)` — specify a proxy server and port.
		`Auto (URL)` — use a proxy configuration file.
		`System` (default) — use the proxy defined for the network profile under **Network > Advanced**.
Proxy: port or URL	Text field	The proxy server and port, or the proxy configuration file URL. Only used with `Manual` or `Auto`.
Proxy exception list	Text field	Addresses that bypass the proxy.

> **Note:**

> For the built-in browser, leave **Proxy type** set to `System`.

Browser Prompts

Browsers can interrupt a Citrix sign-in with prompts. When users sign in to a Citrix store, Google Chrome and Microsoft Edge offer to save the password. Before a website opens Citrix Workspace app, the browsers ask for permission, for example when the store detects Citrix Workspace app or when users start a published desktop or application. With prompt suppression, the browsers skip these prompts, so users of kiosk and shared devices aren't interrupted. Prompt suppression is off by default and needs UniconOS 2609 or later.

Prompt suppression works with Google Chrome, Microsoft Edge and Firefox. Chromium isn't supported. The built-in browser doesn't show these prompts.

The table shows how each browser handles the two prompts while prompt suppression is on:

BROWSER	OFFER TO SAVE PASSWORDS	PERMISSION TO OPEN CITRIX WORKSPACE APP
Google Chrome	Doesn't appear on any website. Chrome also stops saving and filling in addresses and credit card data.	Doesn't appear on trusted origins. Other websites still ask "Open xdg-open?"
Microsoft Edge	Doesn't appear on any website, except in an Edge profile that is signed in with a Microsoft account. Edge also stops saving and filling in addresses and credit card data.	Doesn't appear on trusted origins. Other websites still show "This site is trying to open xdg-open."
Firefox	Never appears on eLux, whether prompt suppression is on or off.	Doesn't appear on trusted origins. Other websites still ask "Allow this site to open the receiver link with Citrix Workspace Launcher?"

TRUSTED ORIGINS

The browsers open Citrix Workspace app without asking only on trusted origins. An origin is the address of a website without a path, for example `https://store.example.com`. The device trusts the stores of its **Citrix StoreFront** applications by itself, each for exactly its host, see Citrix StoreFront. Other origins must be in the list of trusted origins that you set in Scout. Add especially these:

- The identity provider, if users sign in to the store through it, for example with Microsoft Entra ID. With such a federated sign-in, the page of the identity provider opens Citrix Workspace app, not the store.
- A store that users open with a **Browser** application instead of a **Citrix StoreFront** application.

If the device has no **Citrix StoreFront** application and the list is empty, prompt suppression only stops the offer to save passwords.

The list of trusted origins follows these rules:

- Separate the origins with a semicolon or a comma.
- Each origin starts with `https://` and has a host name with at least two parts, such as `login.example.com`. A port is optional, for example `https://store.example.com:8443`.
- The device ignores an origin that breaks these rules, for example an `http://` address, a host name without a dot, a wildcard or a space inside the origin. It shows no message, and the other origins still apply.
- In Chrome and Edge, an origin also covers its subdomains, and an origin without a port covers every port. To trust one host only, put a dot before the host name, for example `https://.store.example.com`.
- In Firefox, an origin covers exactly its host and port. If a store doesn't use port 443, add its port to the origin.

> **Important:**
>
> The device also uses the list of trusted origins for Kerberos authentication. Each origin in the list also allows automatic Kerberos sign-in to that host. Add only origins that you trust.

CONFIGURING PROMPT SUPPRESSION

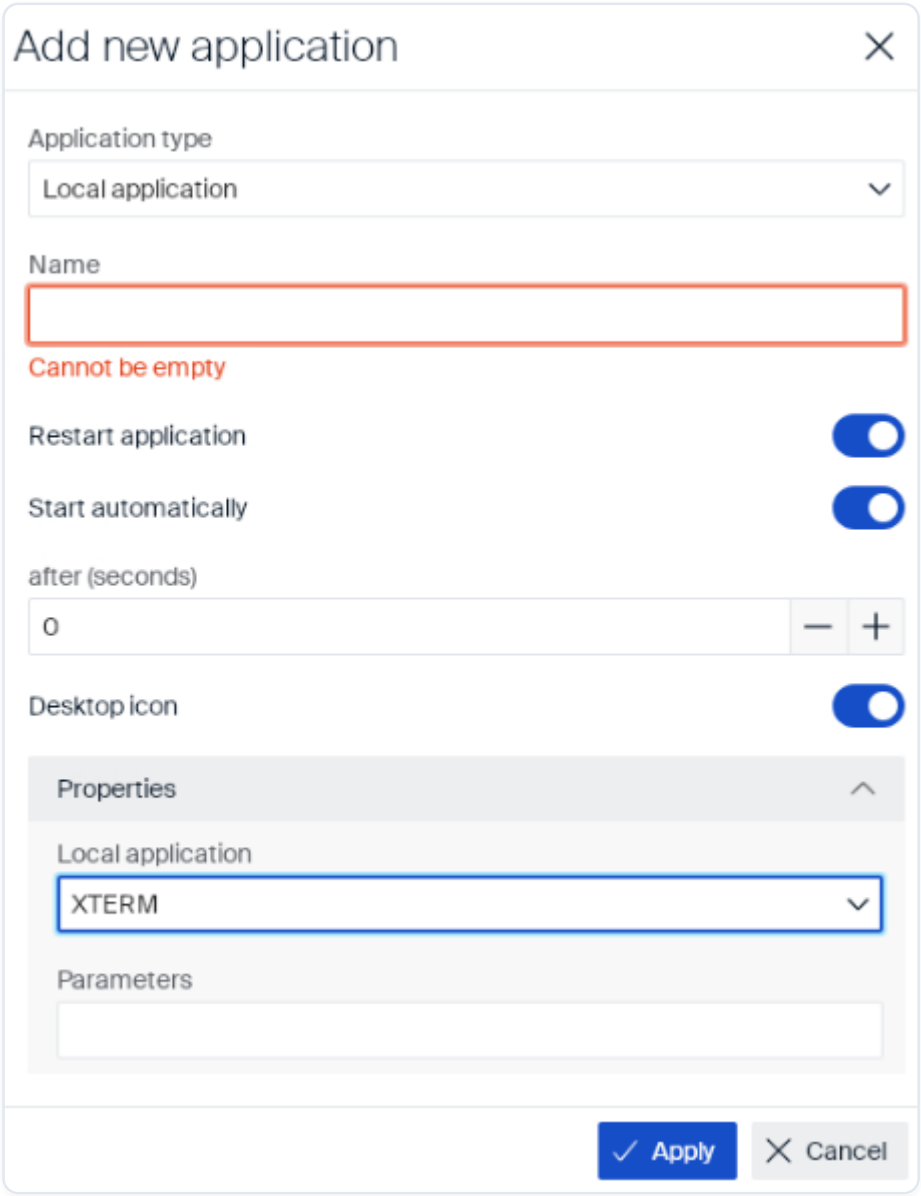
There is no setting for prompt suppression in the device configuration on the device. With UniconOS Management 2611 (Scout 15.2611) and later, you set it in UniconOS Management Board, in the **User authentication** section of the device configuration. **Suppress browser prompts** turns it on, and **Trusted browser origins** holds the list of trusted origins. If **Retain local device configuration (unlocked fields)** is turned on in UniconOS Management Board, the devices apply the list only while the user right **Edit user rights** is locked, see Browser prompts in the **UniconOS Management Board** guide.

With Scout 2609 and earlier, set prompt suppression as advanced file entries instead, one entry per key. In Scout Board, you find advanced file entries in the **Advanced file entries** section of the advanced device configuration. Prompt suppression uses two keys in `/setup/terminal.ini`:

SECTION	KEY	VALUES	DEFAULT	DESCRIPTION
`[Browser]`	`SuppressPrompts`	`true` or `false`	`false`	Only `true` turns on prompt suppression. Any other value, or no entry, leaves the prompts on.
`[Security]`	`BrowserTrustedUris`	List of origins	Empty	The list of trusted origins, for example `https://idp.example.com;https://sso.example.com`.

The browsers apply a change the next time they start, not when the device gets the configuration. A browser that is already running keeps its previous behavior. To apply a change, close all windows of the browser and start it again, or restart the device.

Local Application

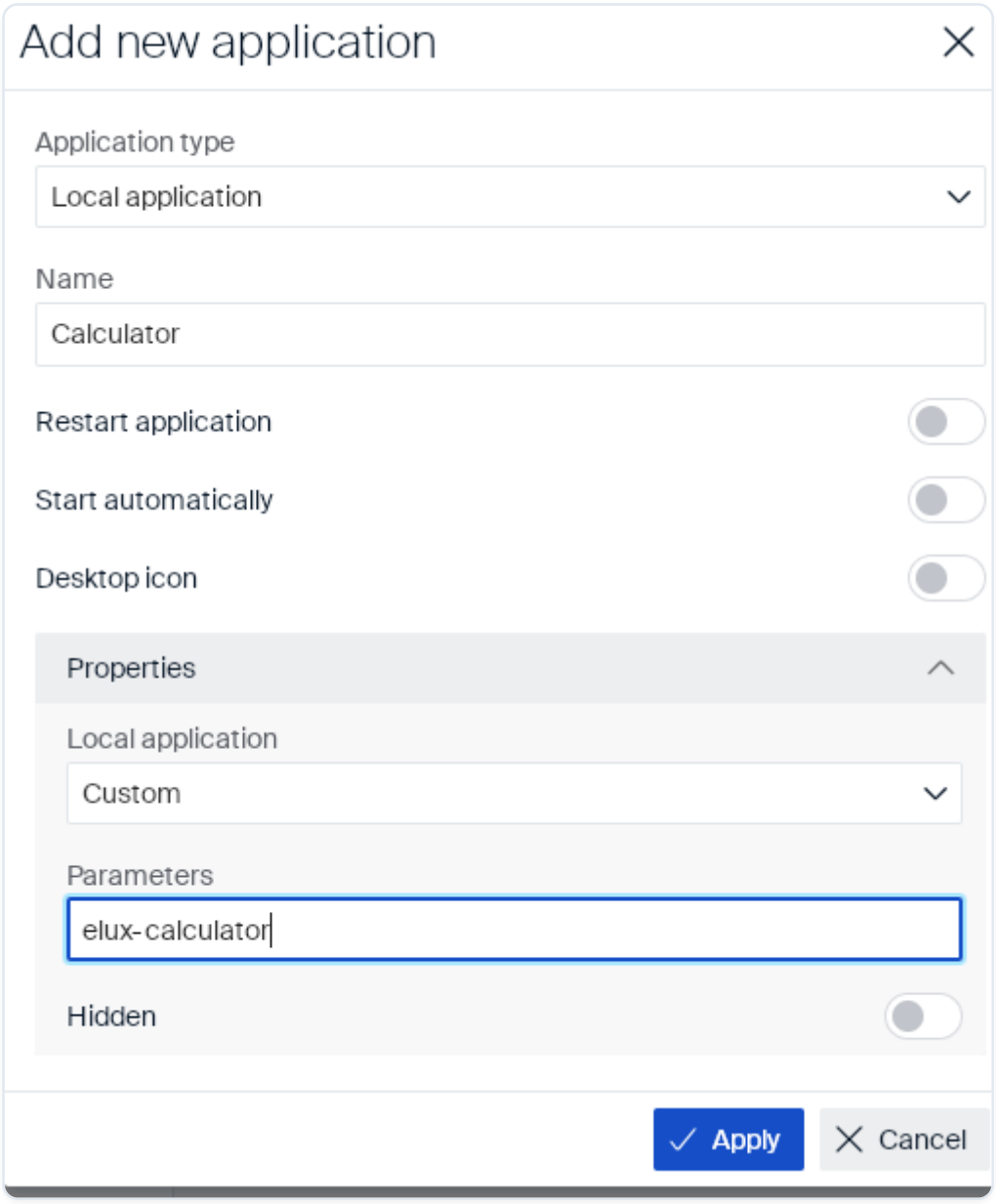


A **Local application** launches a program that's already installed on the device. Besides the common fields, you're presented with the following fields:

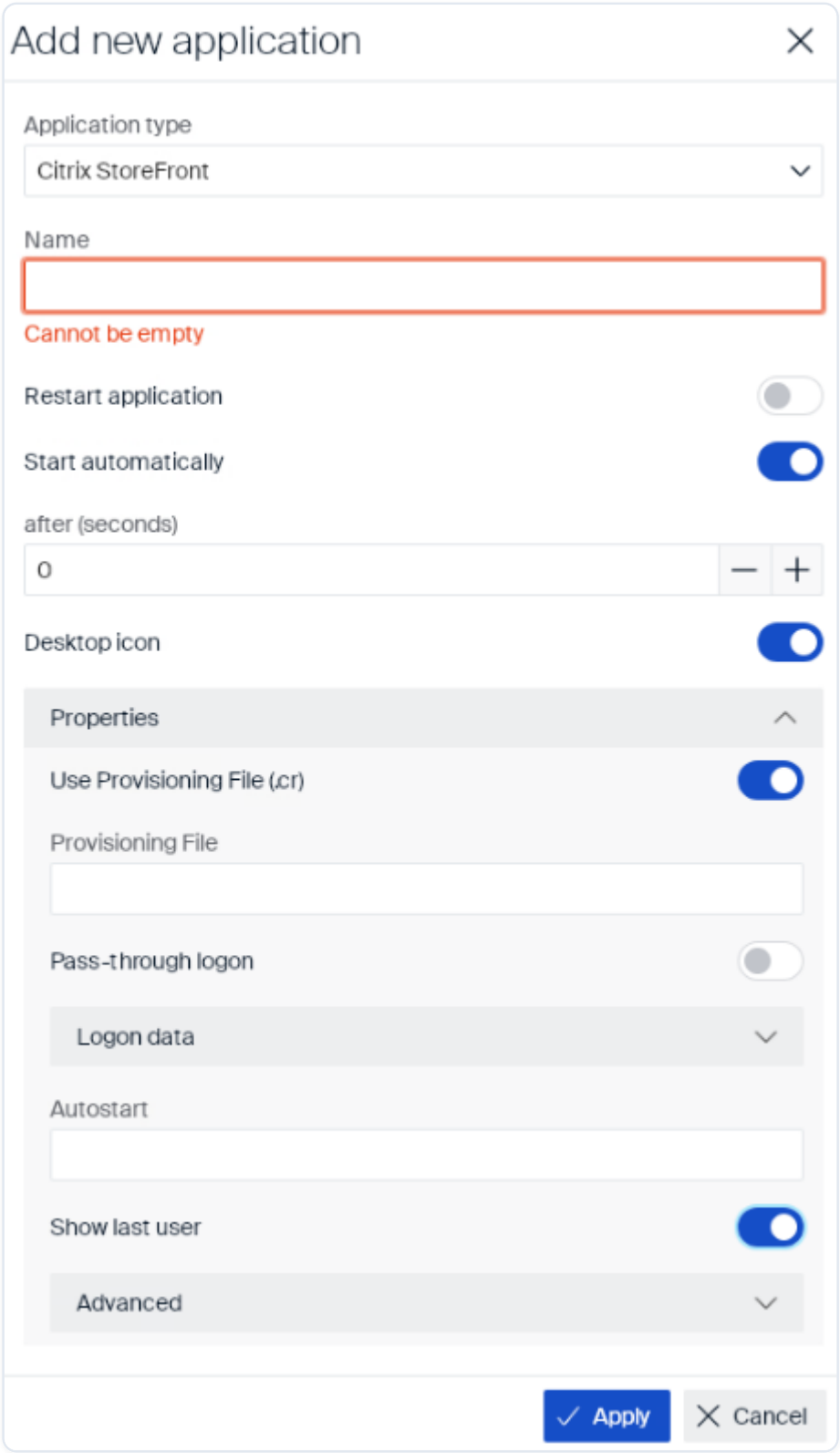
FIELD	TYPE	DESCRIPTION
Local application	Dropdown	`XTERM`, `Resource display`, `File manager`, `Text editor`, `Magnifier`, or `Custom` to start a specific program via **Parameters**. A predefined entry is listed only if the application is installed on the device.

FIELD	TYPE	DESCRIPTION
Parameters	Text field	For a predefined application, additional command-line parameters. For `Custom`, the program name to start, plus any parameters.
Hidden	Toggle	Hides the application from the desktop. Use with **Start automatically** for a background application. Only shown when **Local application** is `Custom`.

Example — a `Custom` local application named **Calculator**, started via **Parameters** `elux-calculator` (the eLux-owned command name, which won't change in future releases):


Add new application
Application type
Local application
Name
Calculator
Restart application
Start automatically
Desktop icon
Properties
Local application
Custom
Parameters
elux-calculator
Hidden
Apply
Cancel

Citrix StoreFront



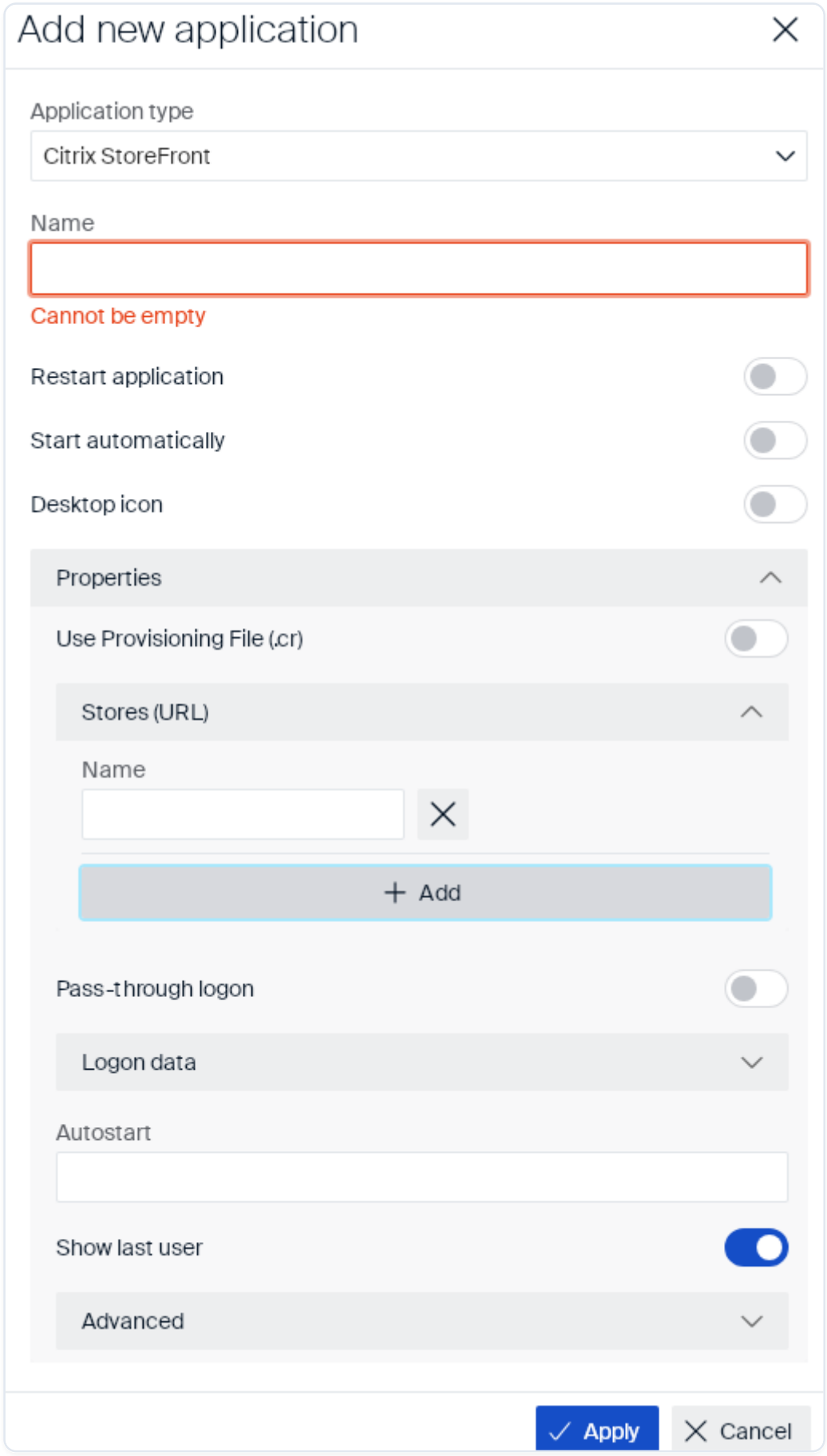
A **Citrix StoreFront** application connects the user to Citrix-published desktops and applications through a store.

Besides the common fields, you're presented with the following fields:

FIELD	TYPE	DESCRIPTION
Use Provisioning File (.cr)	Toggle	Use a StoreFront provisioning file instead of specifying store URLs directly.
Provisioning File	Text field	Path to the provisioning file. Only shown when **Use Provisioning File** is on.
Stores (URL)	Collapsible list	One or more StoreFront store URLs to connect to. Only shown when **Use Provisioning File** is off — see below.
Pass-through logon	Toggle	Logs the user in to the store via single sign-on, using the AD credentials from the eLux desktop login.
Logon data	Collapsible section	Fixed credentials for automatic store logon, used instead of pass-through logon — see below.
Autostart	Text field	Names of the StoreFront resources to start automatically, separated by semicolons — the names must match exactly as shown in StoreFront.
Show last user	Toggle	Shows the last signed-in user's name (not the password) in the logon dialog.

STORES (URL)

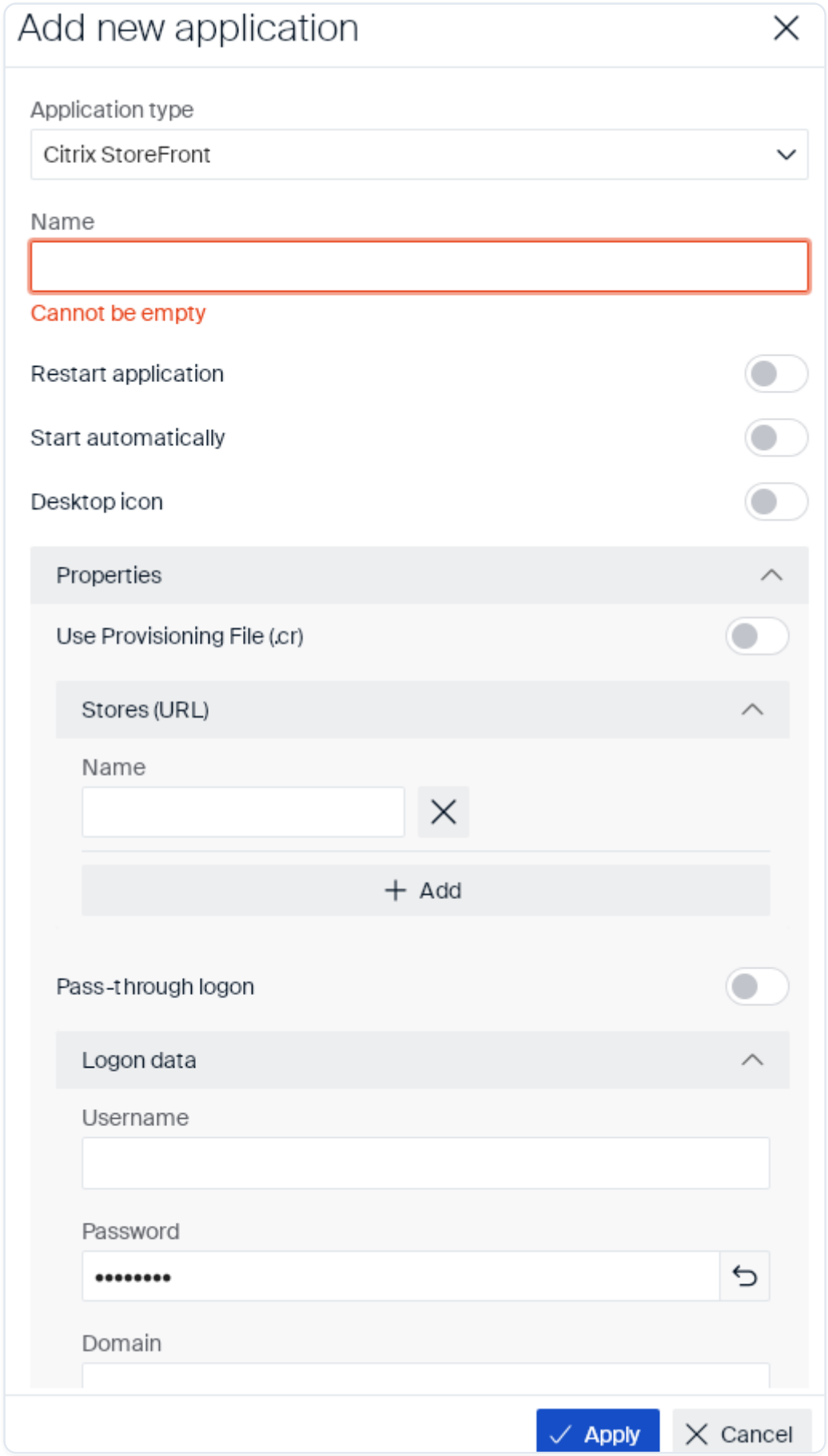
When **Use Provisioning File** is off, **Stores (URL)** lets you list the StoreFront store URLs to connect to directly, instead of using a provisioning file:


Add new application
Application type
Citrix StoreFront
Name
Cannot be empty
Restart application
Start automatically
Desktop icon
Properties
Use Provisioning File (.cr)
Stores (URL)
Name
Add
Pass-through logon
Logon data
Autostart
Show last user
Advanced
Apply
Cancel

FIELD	TYPE	DESCRIPTION
Name	Text field	URL of a StoreFront store. Click **+ Add** for another store, or the **X** to remove one.

LOGON DATA

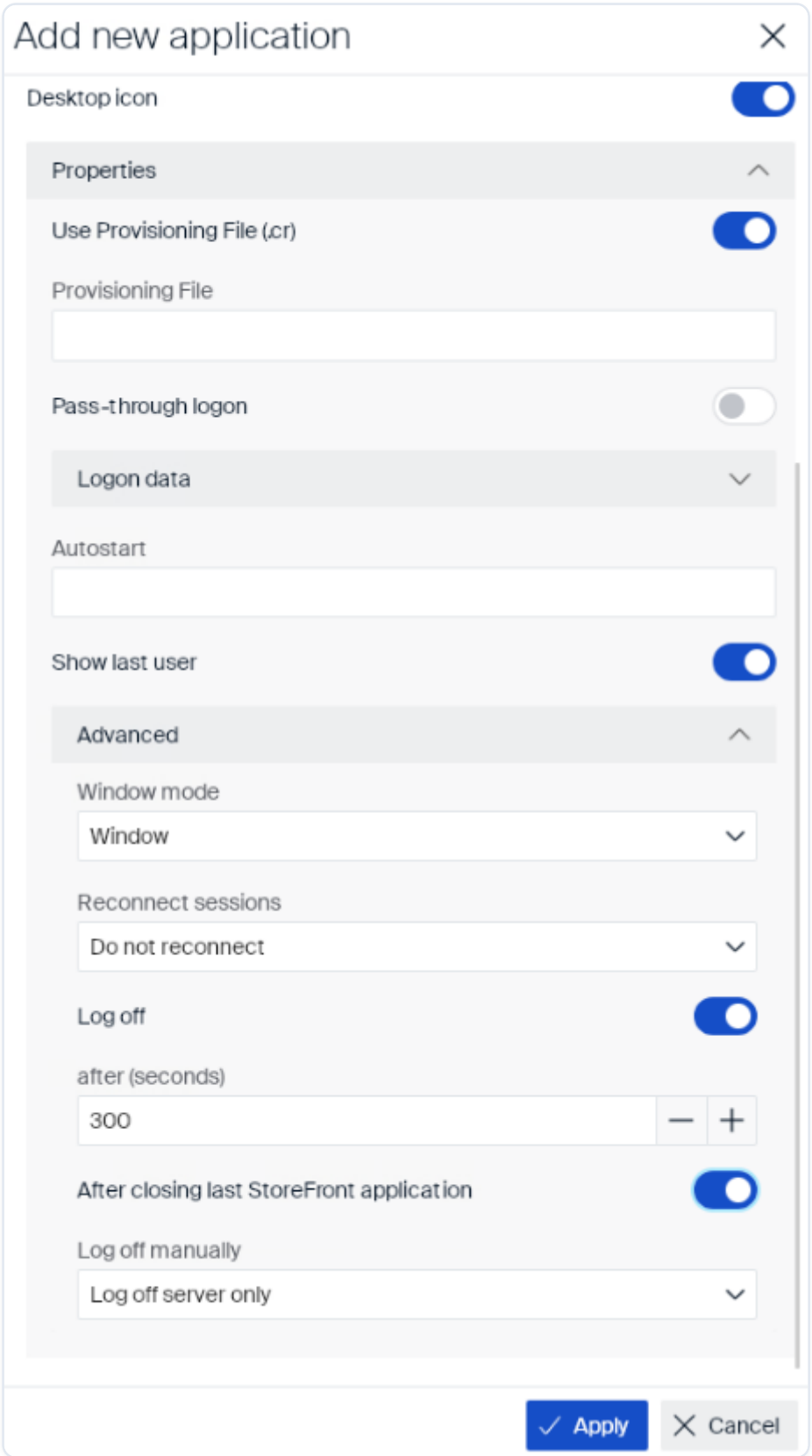
Logon data lets you provide fixed credentials for automatic sign-in, used instead of pass-through logon:


Add new application
Application type
Citrix StoreFront
Name
Cannot be empty
Restart application
Start automatically
Desktop icon
Properties
Use Provisioning File (.cr)
Stores (URL)
Name
Add
Pass-through logon
Logon data
Username
Password
Domain
Apply
Cancel

FIELD	TYPE	DESCRIPTION
Username	Text field	User name for automatic logon (used instead of pass-through logon).
Password	Text field	Password for automatic logon.
Domain	Text field	Domain for automatic logon.

ADVANCED

The **Advanced** section controls how a StoreFront session behaves once it's running — whether it opens in a window or full screen, what happens on reconnect, and how the user is logged off:

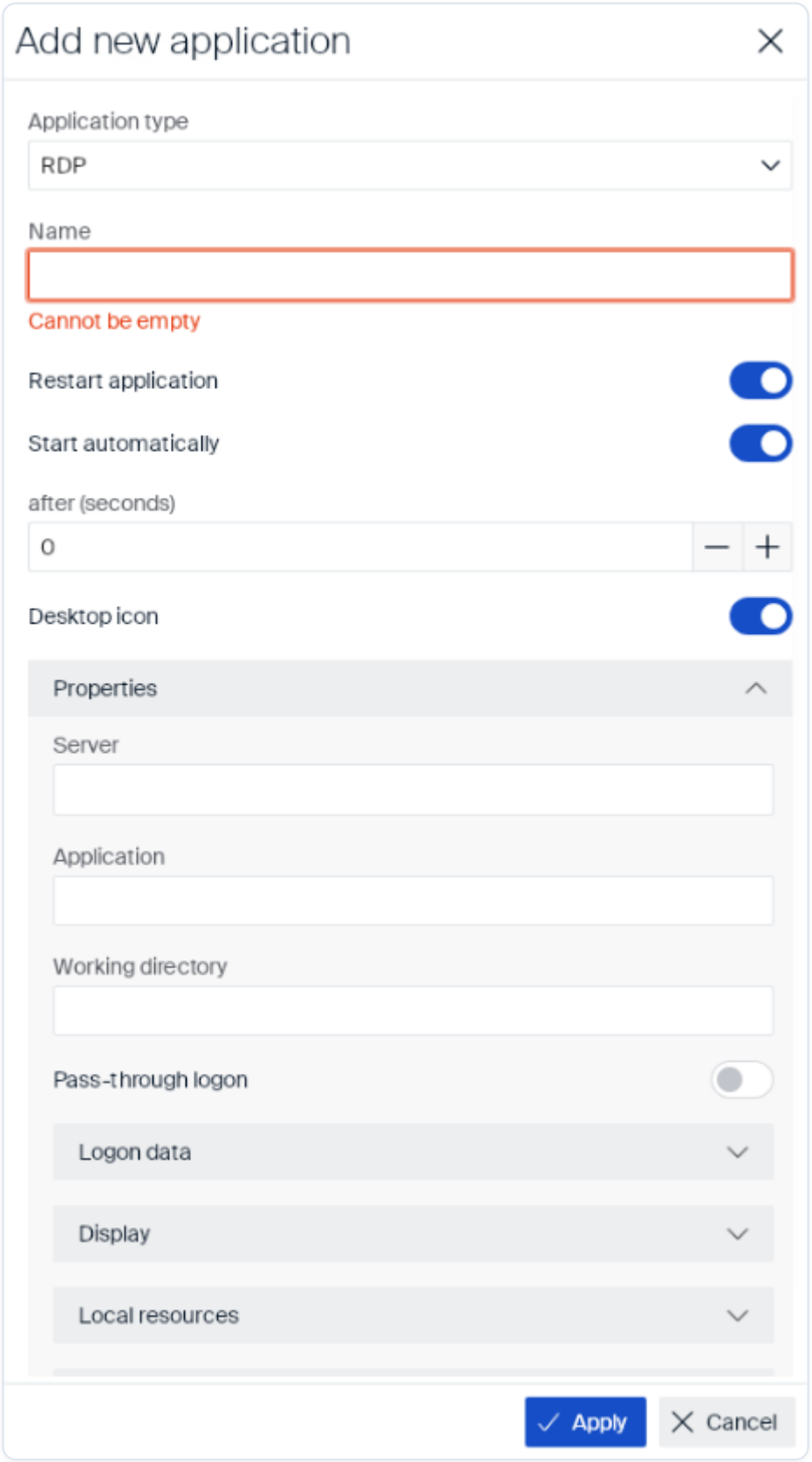


FIELD	TYPE	DESCRIPTION
Window mode	Dropdown	`Window` or `Full screen`.

FIELD	TYPE	DESCRIPTION
Reconnect sessions	Dropdown	`Do not reconnect` (default) — the connection to a desktop or published application isn't restored on reconnect.
		`Disconnected sessions only` — restores a disconnected session.
		`Active and disconnected sessions` — restores either an active or a disconnected session.
Log off	Toggle	Automatically logs off from the StoreFront server after a delay. Doesn't affect the launched desktop.
after (seconds)	Number field	Delay before automatic log off, in seconds. Default: `300`. Only shown when **Log off** is on.
After closing last StoreFront application	Toggle	Logs off automatically once the last open StoreFront resource is closed, instead of waiting for the delay.
Log off manually	Dropdown	`Log off server only` (default) — logs off only from the StoreFront server.
		`Log off server and applications` — also logs off the virtual desktop or published applications.
		`Log off server and disconnect applications`

FIELD	TYPE	DESCRIPTION
		— logs off the server but only disconnects the virtual desktop or published applications, so the user can reconnect later.

RDP

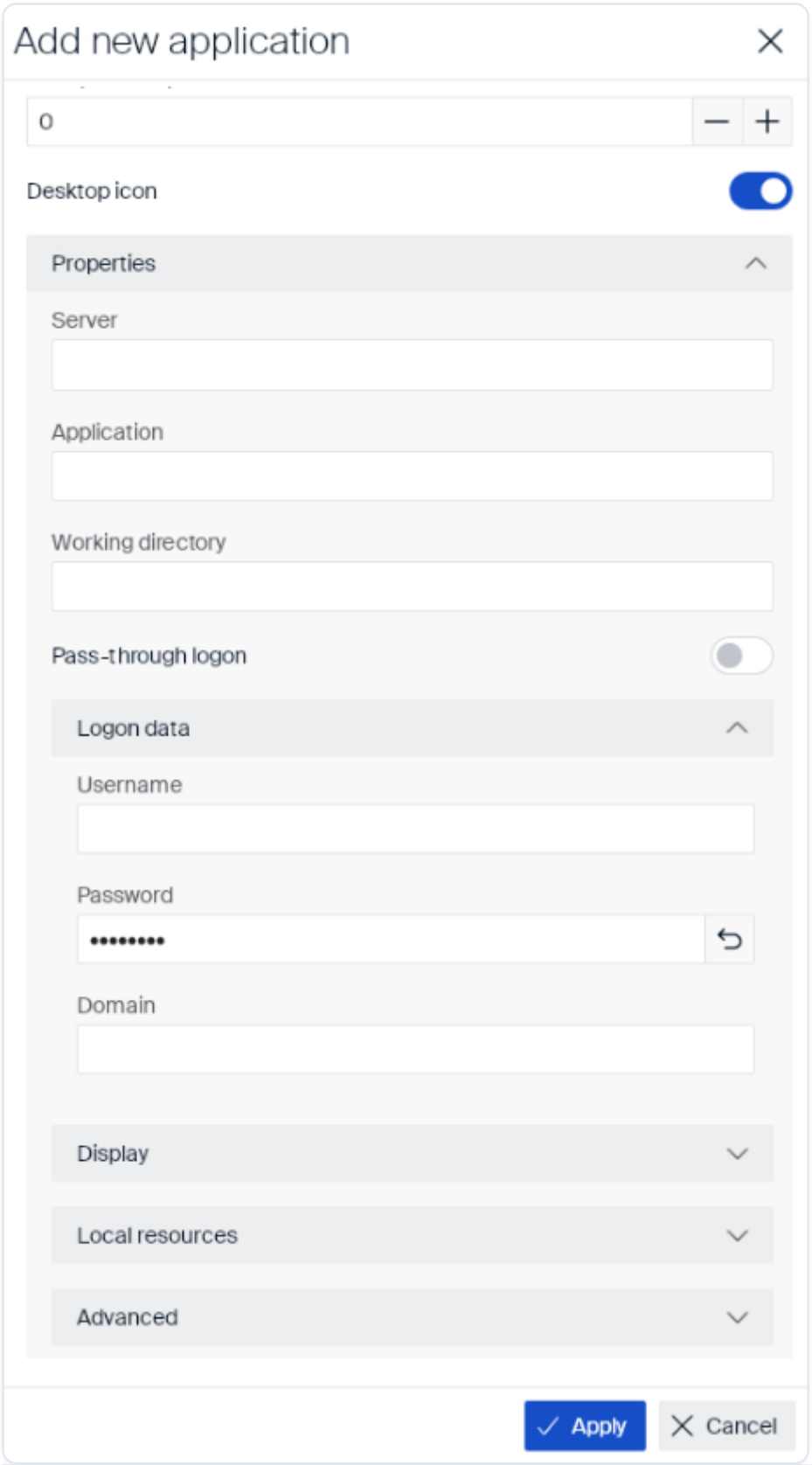


An **RDP** application opens a remote Windows desktop, or a specific Windows program on a server. Besides the common fields, you're presented with the following fields:

FIELD	TYPE	DESCRIPTION
Server	Text field	Name or address of the RDP server.
Application	Text field	Path of the application to start on the server. Leave empty to start a full desktop.
Working directory	Text field	Working directory for the application on the server.
Pass-through logon	Toggle	Logs the user in using the AD credentials from the eLux desktop login.

LOGON DATA

Logon data lets you provide fixed credentials for automatic sign-in, used instead of pass-through logon:

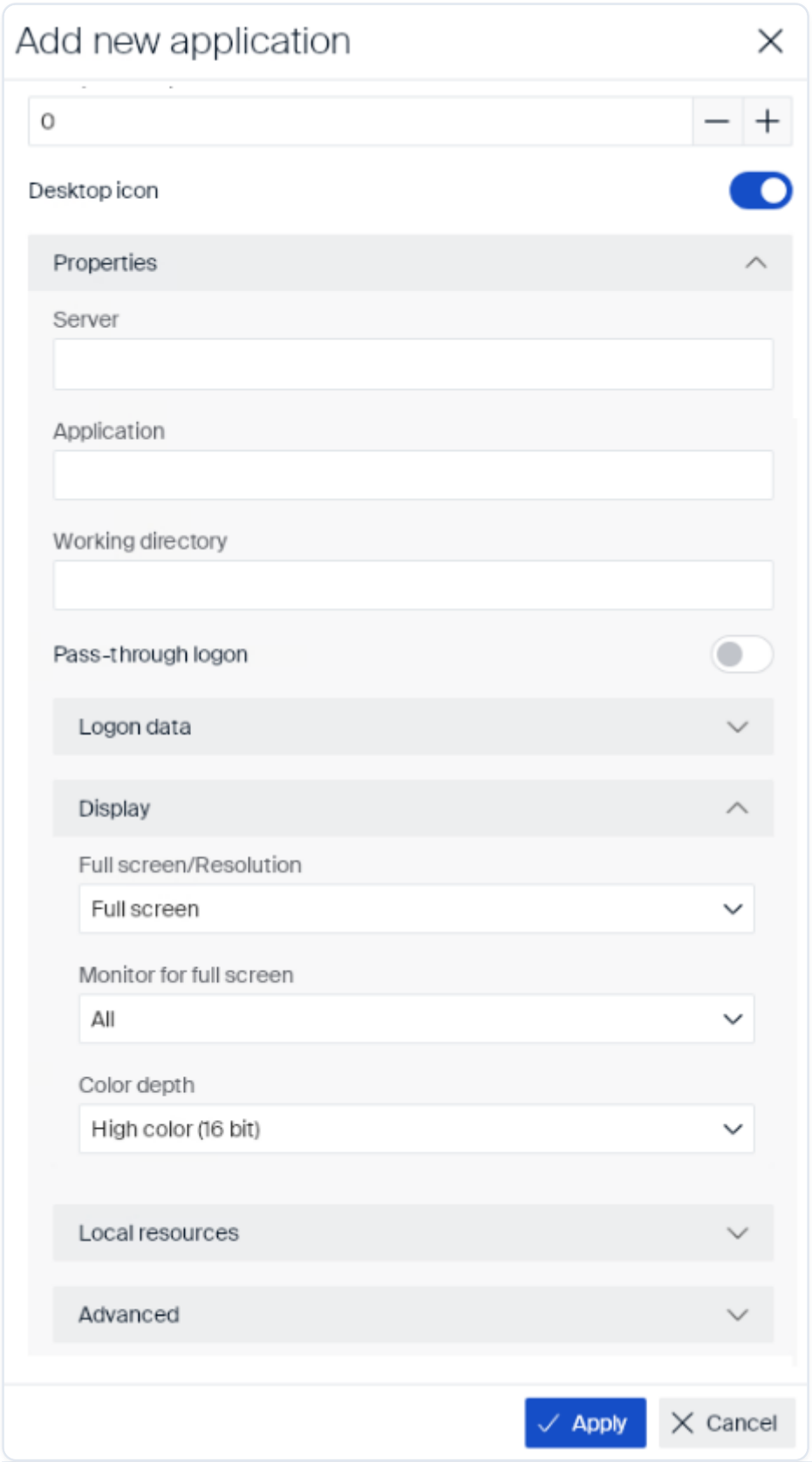


FIELD	TYPE	DESCRIPTION
Username	Text field	User name for automatic logon (used instead of pass-through logon).

FIELD	TYPE	DESCRIPTION
Password	Text field	Password for automatic logon.
Domain	Text field	Domain for automatic logon.

DISPLAY

Display settings control how the session window appears — its size, which monitor it uses, and the color quality:

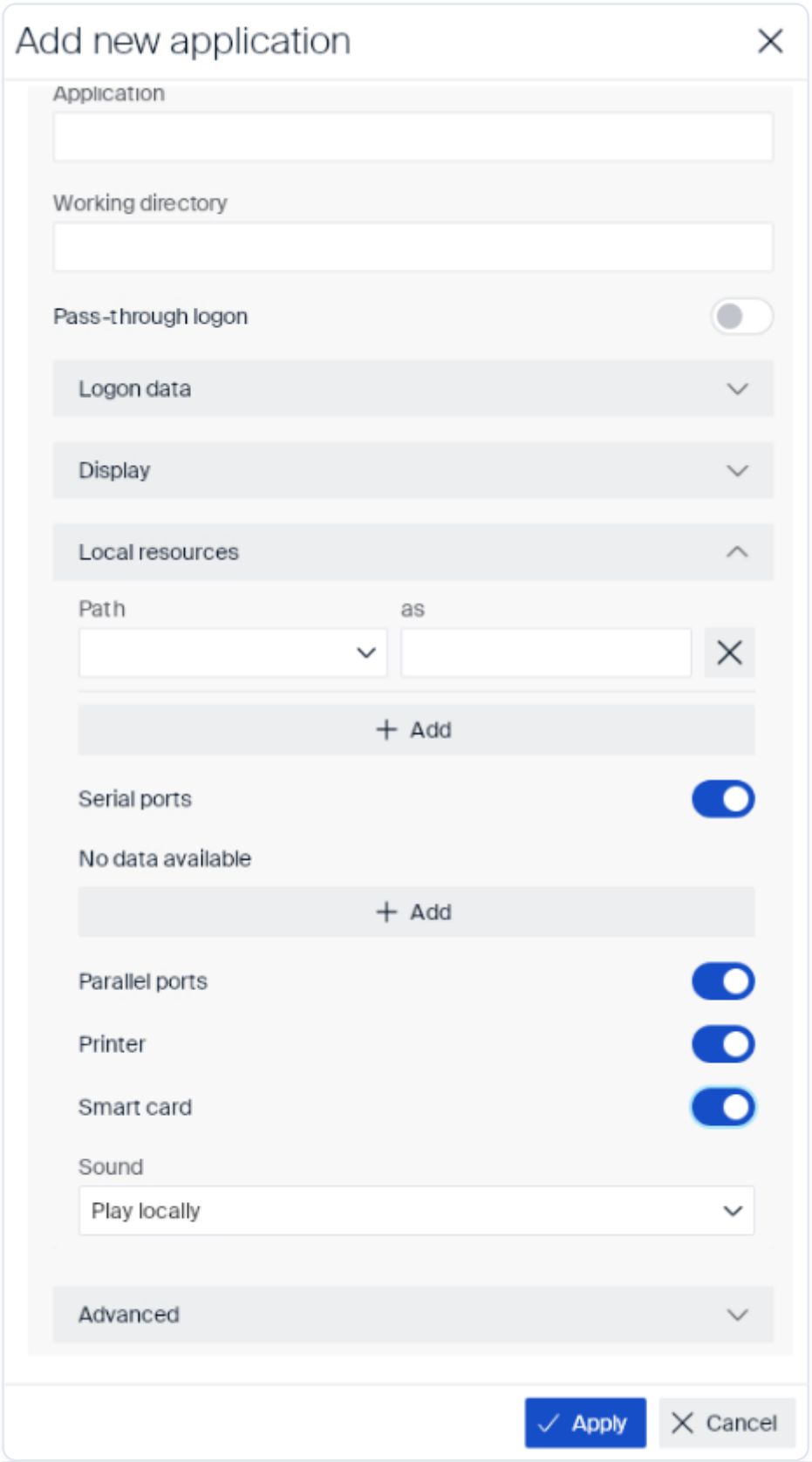


FIELD	TYPE	DESCRIPTION
Full screen/Resolution	Dropdown	`Full screen`, or a specific resolution: `640x480`,

FIELD	TYPE	DESCRIPTION
		`800x600`, `1024x768`, `1152x864`, or `1280x1024`.
Monitor for full screen	Dropdown	`All`, or a specific connected monitor. Only applies when **Full screen/ Resolution** is `Full screen`.
Color depth	Dropdown	`256 (8 bit)`, `High color (16 bit)` (default), `True color (24 bit)`, or `True color (32 bit)`.

LOCAL RESOURCES

Local resources settings decide which of the device's local drives, ports, and peripherals are shared into the session:

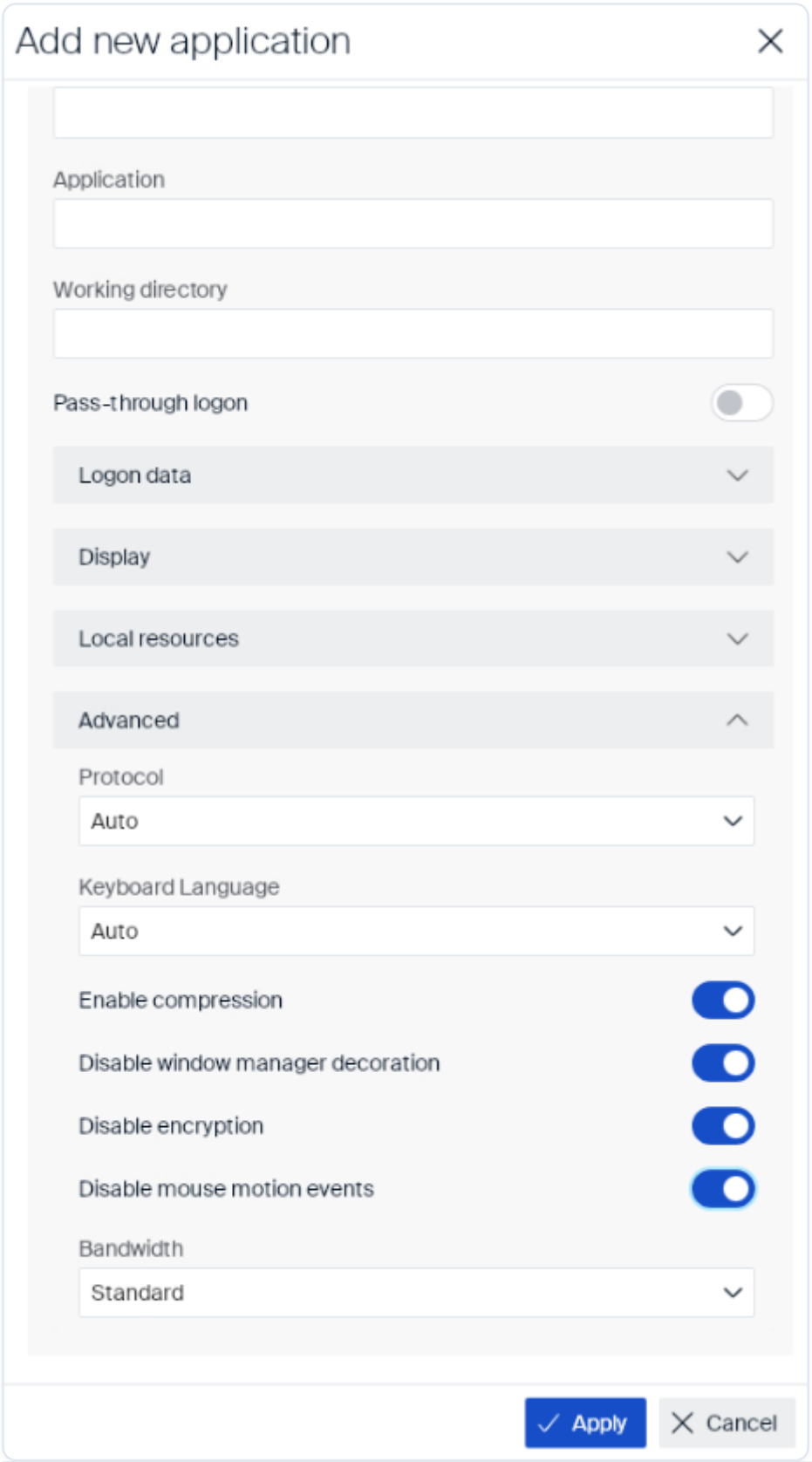


FIELD	TYPE	DESCRIPTION
Path / as	Dropdown + text field	Maps a local drive or folder into the session. Click **+ Add** for another mapping.

FIELD	TYPE	DESCRIPTION
Serial ports	Toggle	Makes local serial (COM) ports available in the session.
COM port / as	Dropdown + number field	Only shown after turning on **Serial ports** and clicking **+ Add**. Maps a local COM port to a port number in the session.
Parallel ports	Toggle	Makes local parallel ports available in the session.
Printer	Toggle	Makes locally connected printers available in the session.
Smart card	Toggle	Allows certificate-based logon using a smart card.
Sound	Dropdown	`Off` — session sound is disabled. `Play locally` (default) — plays session sound on the device. `Play remotely` — plays session sound on the server.

ADVANCED

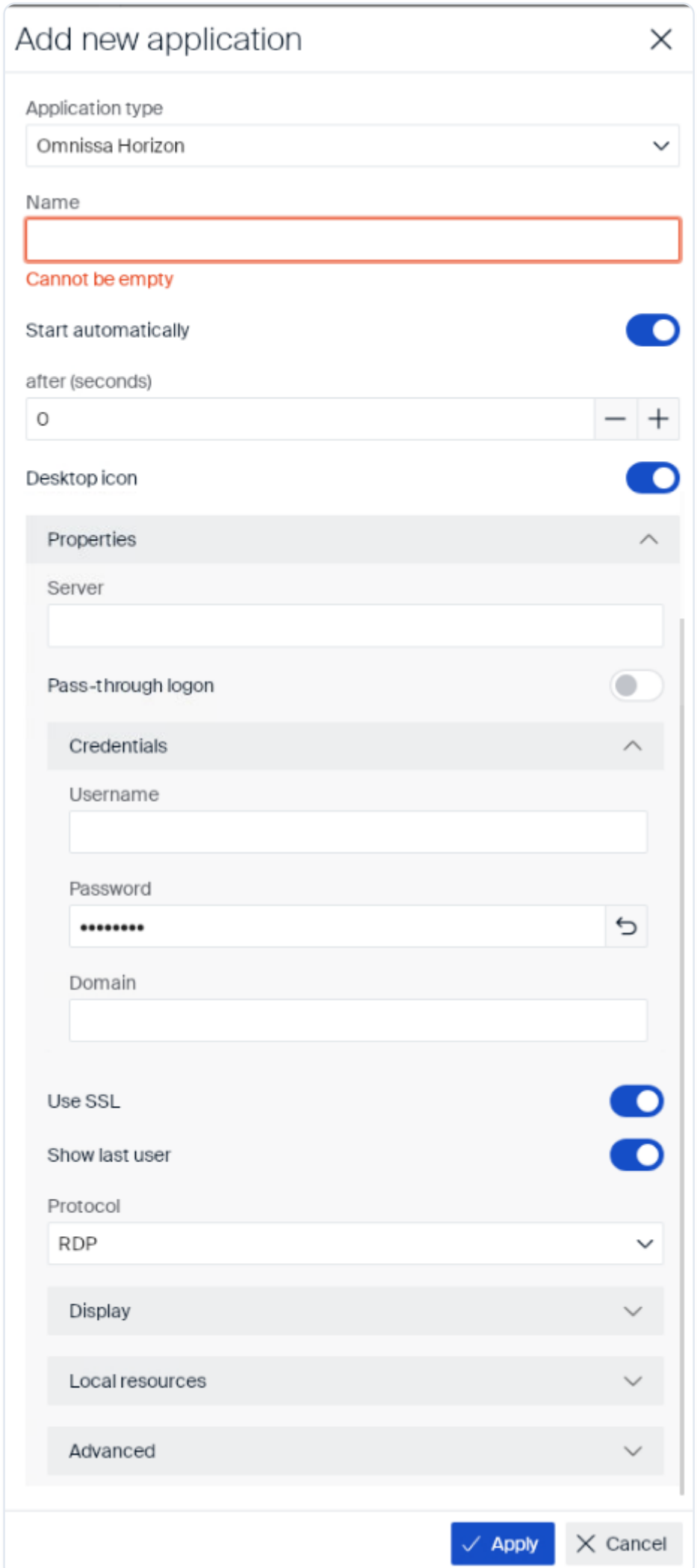
Advanced settings fine-tune the RDP protocol itself — connection version, keyboard layout, and performance trade-offs for slower connections:



FIELD	TYPE	DESCRIPTION
Protocol	Dropdown	`Auto` (default) — the RDP protocol version is detected automatically. `RDP V4` or

FIELD	TYPE	DESCRIPTION
		`RDP V5` — force a specific version.
Keyboard Language	Dropdown	`Auto` (default) — matches the device's **Keyboard** configuration, or a specific language.
Enable compression	Toggle	Compresses session data to reduce bandwidth use.
Disable window manager decoration	Toggle	Hides the window frame around the session.
Disable encryption	Toggle	Disables session encryption. Can improve performance; disabled by default for security.
Disable mouse motion events	Toggle	Sends mouse position only on click, not continuously. Reduces bandwidth use, useful on slow connections.
Bandwidth	Dropdown	`Standard`, `Modem`, `Broadband`, or `LAN`.

Omnissa Horizon

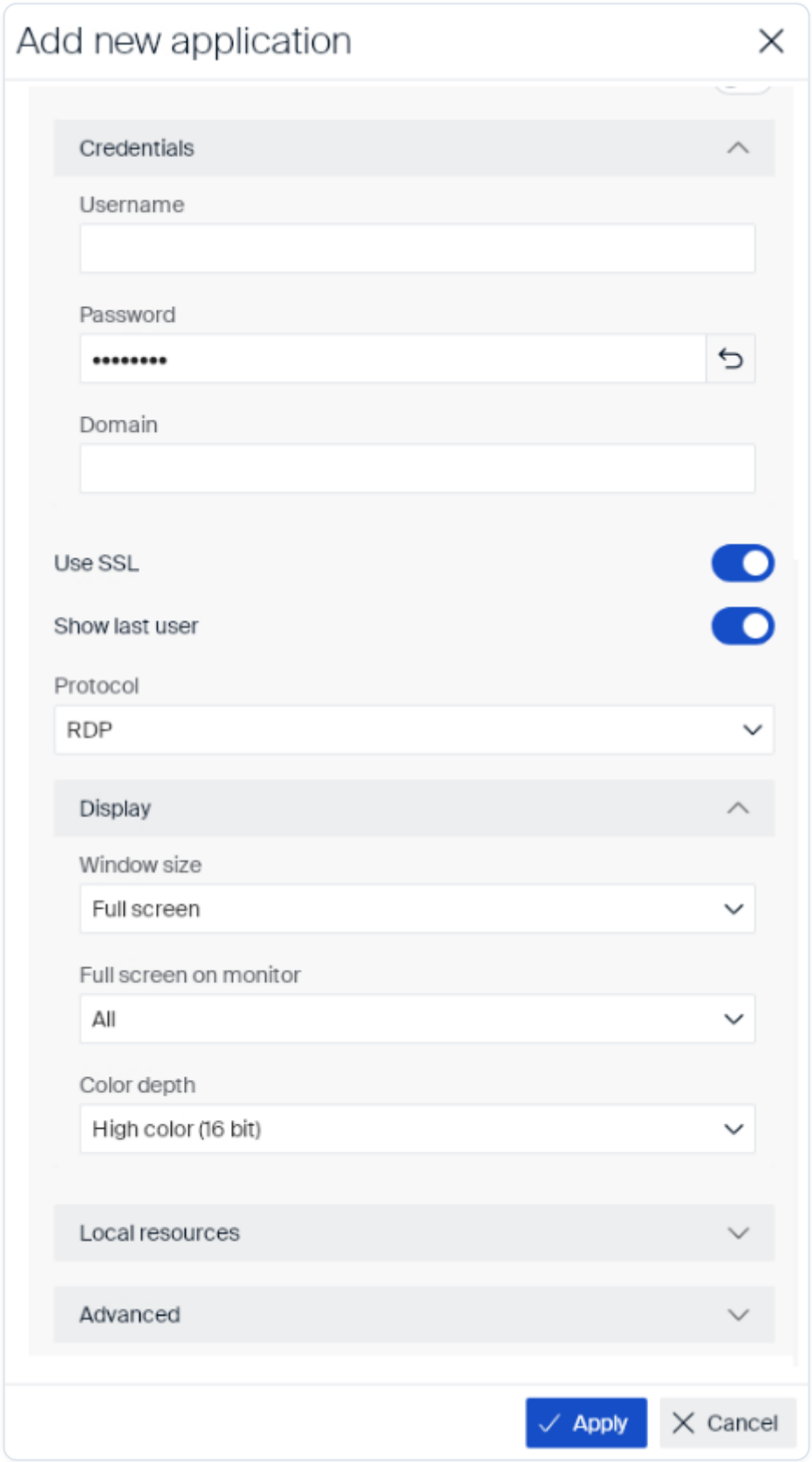

Add new application
Application type
Omnissa Horizon
Name
Cannot be empty
Start automatically
after (seconds)
0
Desktop icon
Properties
Server
Pass-through logon
Credentials
Username
Password
Domain
Use SSL
Show last user
Protocol
RDP
Display
Local resources
Advanced
Apply
Cancel

An **Omnissa Horizon** application connects to a VDI desktop through Omnissa Horizon (formerly VMware Horizon). Besides the common fields, you're presented with the following fields:

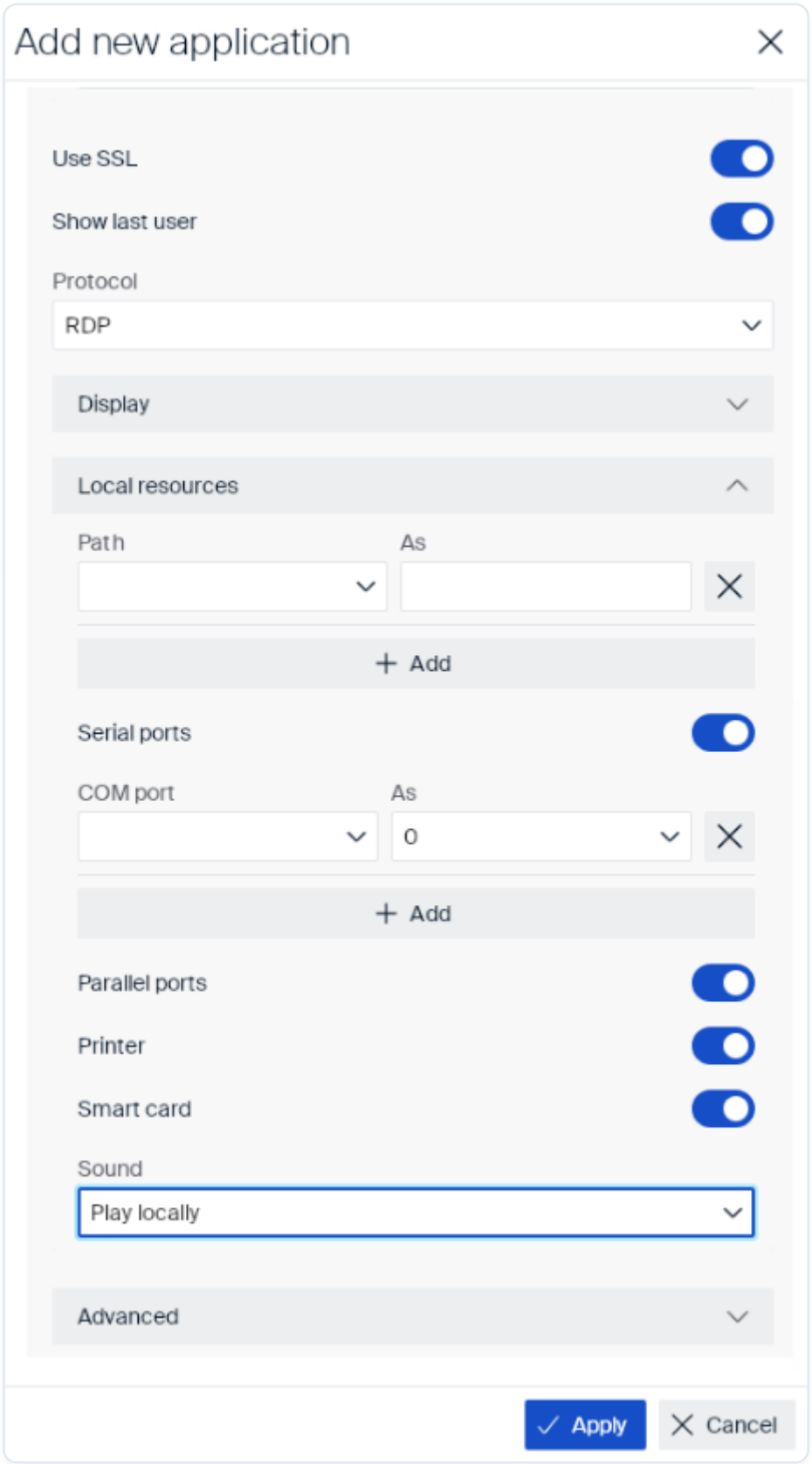
FIELD	TYPE	DESCRIPTION
Server	Text field	Name or address of the Omnissa Horizon server.
Pass-through logon	Toggle	Logs the user in using the AD credentials from the eLux desktop login.
Username	Text field	User name for automatic logon (used instead of pass-through logon), under **Credentials**.
Password	Text field	Password for automatic logon, under **Credentials**.
Domain	Text field	Domain for automatic logon, under **Credentials**.
Use SSL	Toggle	Connects to the server over SSL/TLS.
Show last user	Toggle	Shows the last signed-in user's name (not the password) in the logon dialog.
Protocol	Dropdown	`RDP` (default), `PCoIP`, or `Omnissa Blast`.

Display, **Local resources**, and **Advanced** use the same fields as RDP, with these differences:

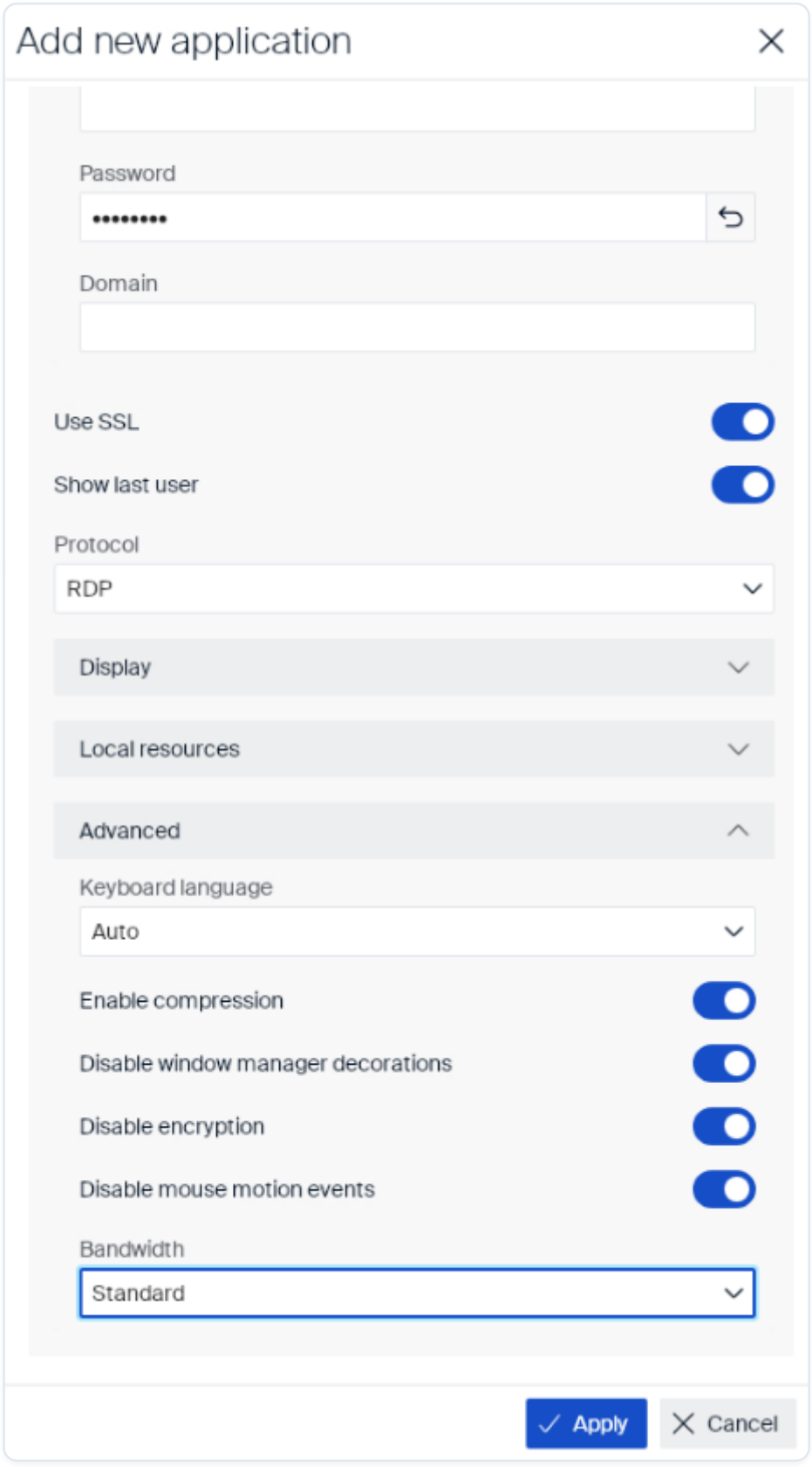
- **Display**: fields are named **Window size**, **Full screen on monitor**, and **Color depth** — same meaning as RDP's equivalents.



- **Local resources**: identical to RDP.



- **Advanced**: identical to RDP, except there is no separate **Protocol** field here (Protocol is set under Properties, described above).

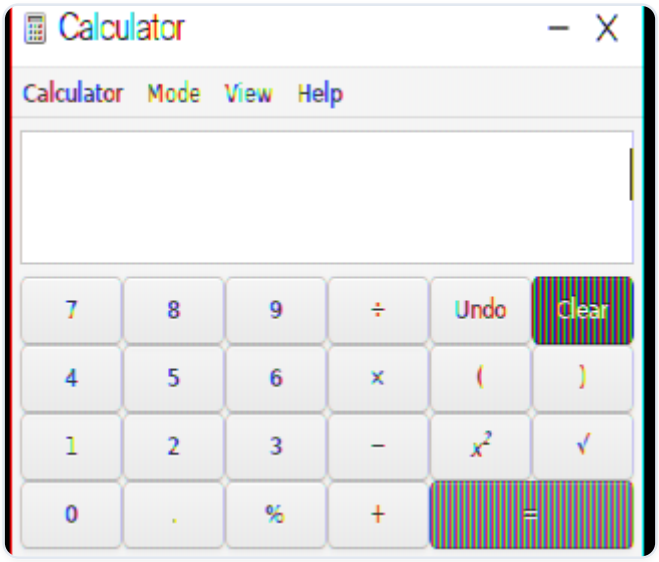


New Applications

eLux now ships a modern calculator and a modern screenshot tool.

CALCULATOR

The bundled calculator is now **MATE Calculator**. It shows as **Calculator** everywhere it appears — the app menu, the window title bar, and the taskbar — and it's actively maintained, unlike the *galculator* tool it replaces.



To make the calculator available on a device, enable **calc** under **Packages → FPM** in the desktop_environment EPM.

You can start it in two ways:

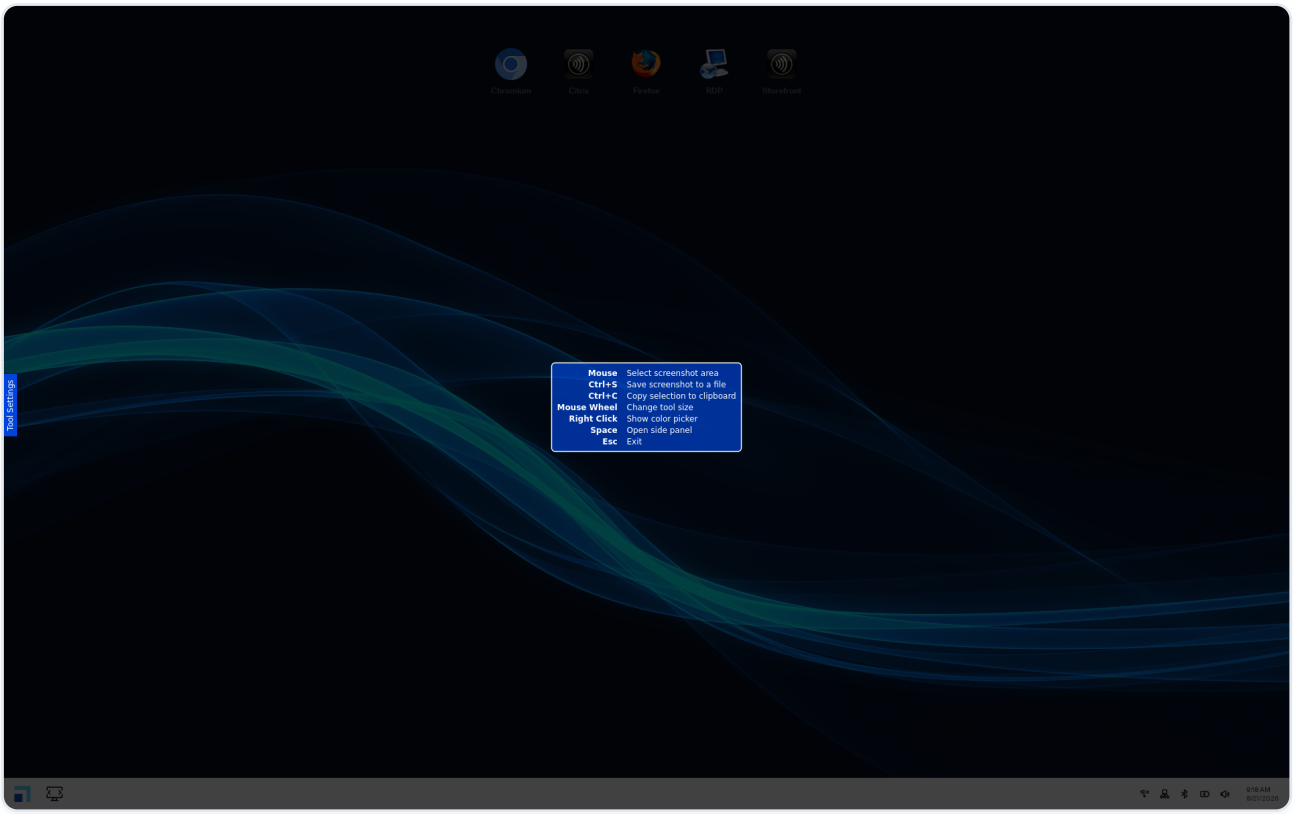
COMMAND	NOTES
`elux-calculator`	Recommended — the eLux-owned command name, which won't change in future releases.
`galculator`	Still works, for existing local application definitions.

> **Note:**

> When adding the calculator as a Local Application, use `elux-calculator` as shown in the example above.

SCREENSHOT

The bundled screenshot tool is now **Flameshot**. You can capture a region or the full screen, annotate it with arrows, text, blur, and highlight, then save it or paste it straight into another application as a real image.



Screenshots save to **/home/elux/Screenshots**. This folder is recreated automatically if you delete it.

To make the screenshot tool available on a device, enable both of the following under **Packages** → **FPM** in the desktop_environment EPM:

- **screenshot** — ships the tool itself.
- **QT** — a dependency required by Flameshot.

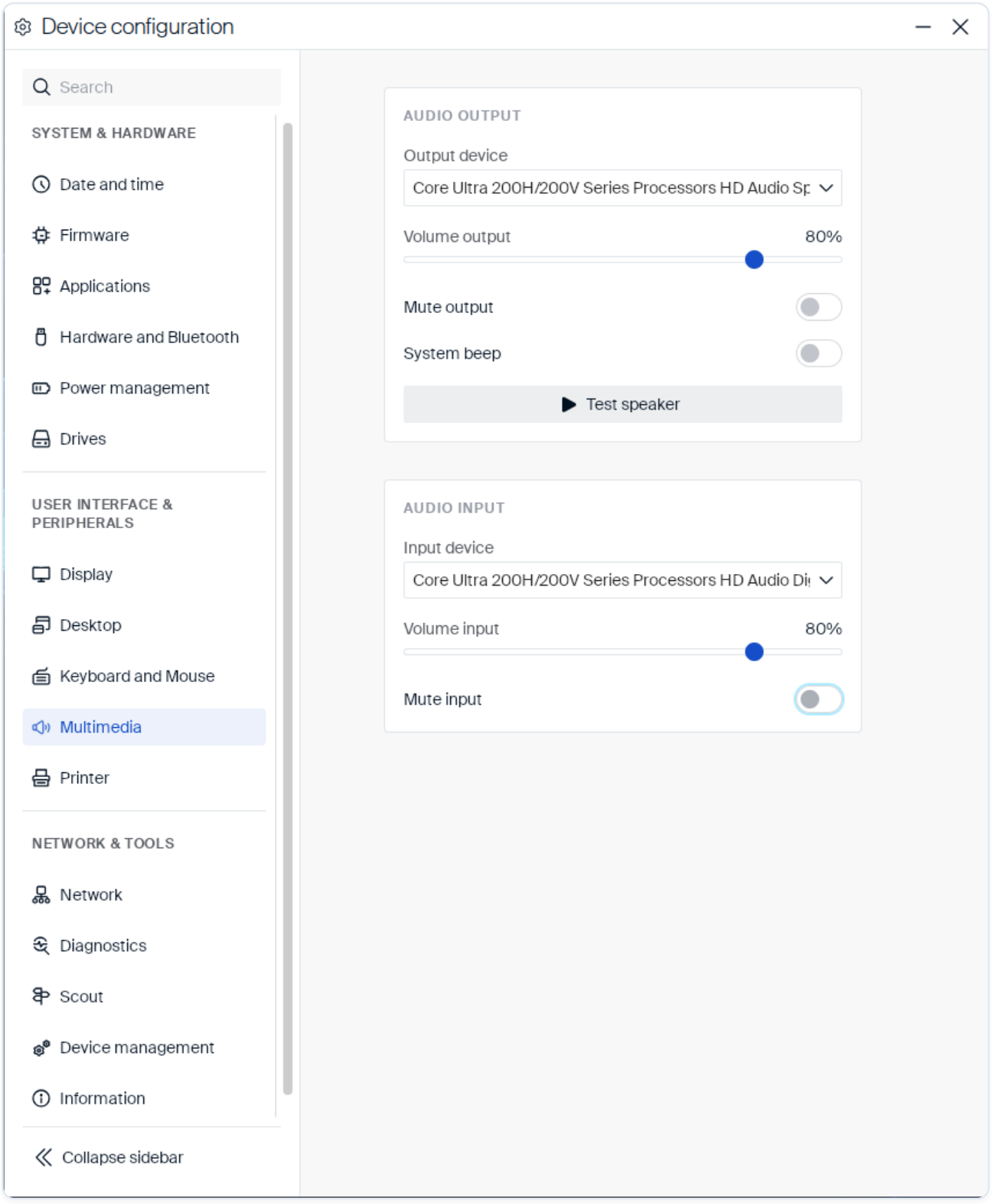
You can trigger a capture in several ways:

SHORTCUT OR COMMAND	WHAT IT DOES
Print	Opens the capture UI — drag a region, annotate, then save or copy it.
Ctrl + Print	Captures the full screen straight to the clipboard.
Super + Shift + S	Opens the capture UI.
`elux-screenshot` (terminal)	Opens the capture UI.
`flameshot full -p /home/elux/Screenshots` (terminal)	Saves a full-screen capture directly to the Screenshots folder.

The legacy `mate-screenshot` command still works.

Multimedia

Multimedia is where you configure the device's audio output and input, including volume, muting, and the system beep.



Audio output

Choose which device plays sound, and control its volume and the system beep:

SETTING	TYPE	DESCRIPTION
Output device	Dropdown	The audio device used for playback. Lists every audio output device eLux® detects, for example a sound card's analog output or a monitor's HDMI audio.
Volume output	Slider	The playback volume level.
Mute output	Toggle	Mutes audio playback. While muted, **Volume output** shows 0%. Turning **Mute output** off restores the actual volume level.
System beep	Toggle	Enables or disables the system beep sound.
Test speaker	Button	Plays a test sound through the selected output device.

Audio input

Choose which device records sound, and control its volume:

SETTING	TYPE	DESCRIPTION
Input device	Dropdown	The audio device used for recording, for example a microphone jack or a headset's built-in microphone.
Volume input	Slider	The recording volume level.
Mute input	Toggle	Mutes audio recording. While muted, **Volume input** shows 0%. Turning **Mute input** off

SETTING	TYPE	DESCRIPTION
		restores the actual volume level.

USB audio auto-accept

When you connect a USB audio device, eLux® shows a **"New audio device detected"** notification with two buttons:

- **Use device** — sets it as the default for this session only; you're asked again next time.
- **Remember device** — sets it as the default now and on every reconnect, with no more prompts.

If you miss the notification, it stays in the Notification Center so you can respond to it later.

Remember device only sets the **default** audio device — other devices remain usable and selectable. It is offered for USB devices only.

Whether **Remember device** survives a reboot depends on the administrator (see the whitelist/blacklist keys below). If persistence across reboots isn't enabled, remembering still applies for the current session.

A device blocked by the administrator (via `IgnoreAsDefault`, below) can never be remembered — the administrator's policy always takes precedence.

An administrator can also control this per device, group, or OU with two keys in `/setup/terminal.ini`, section `[Multimedia]`, set via **Scout → Advanced Configuration → File Entries**:

KEY	EFFECT
`AudioAutoAcceptList`	Whitelist — listed devices become default automatically, no notification.
`IgnoreAsDefault`	Blacklist — listed devices never become default, no notification.

Devices in neither list still show the notification. If a device is in both, the blacklist wins.

Both keys are empty by default, so every new device shows the notification until you add one.

Both take comma-separated `vendorId:productId` pairs (4-digit hex, case-insensitive) — find a device's ID with `lsusb` (for example, `ID 046d:0a4d` → use `046d:0a4d`):

```
[Multimedia]

AudioAutoAcceptList=all              # every device, like eLux RP6
(whitelist only)

AudioAutoAcceptList=0d8c:0014,046d:* # only these: one exact device, and
all from vendor 046d

IgnoreAsDefault=8087:0aaa            # never default this device (e.g. a
dock/monitor speaker)
```

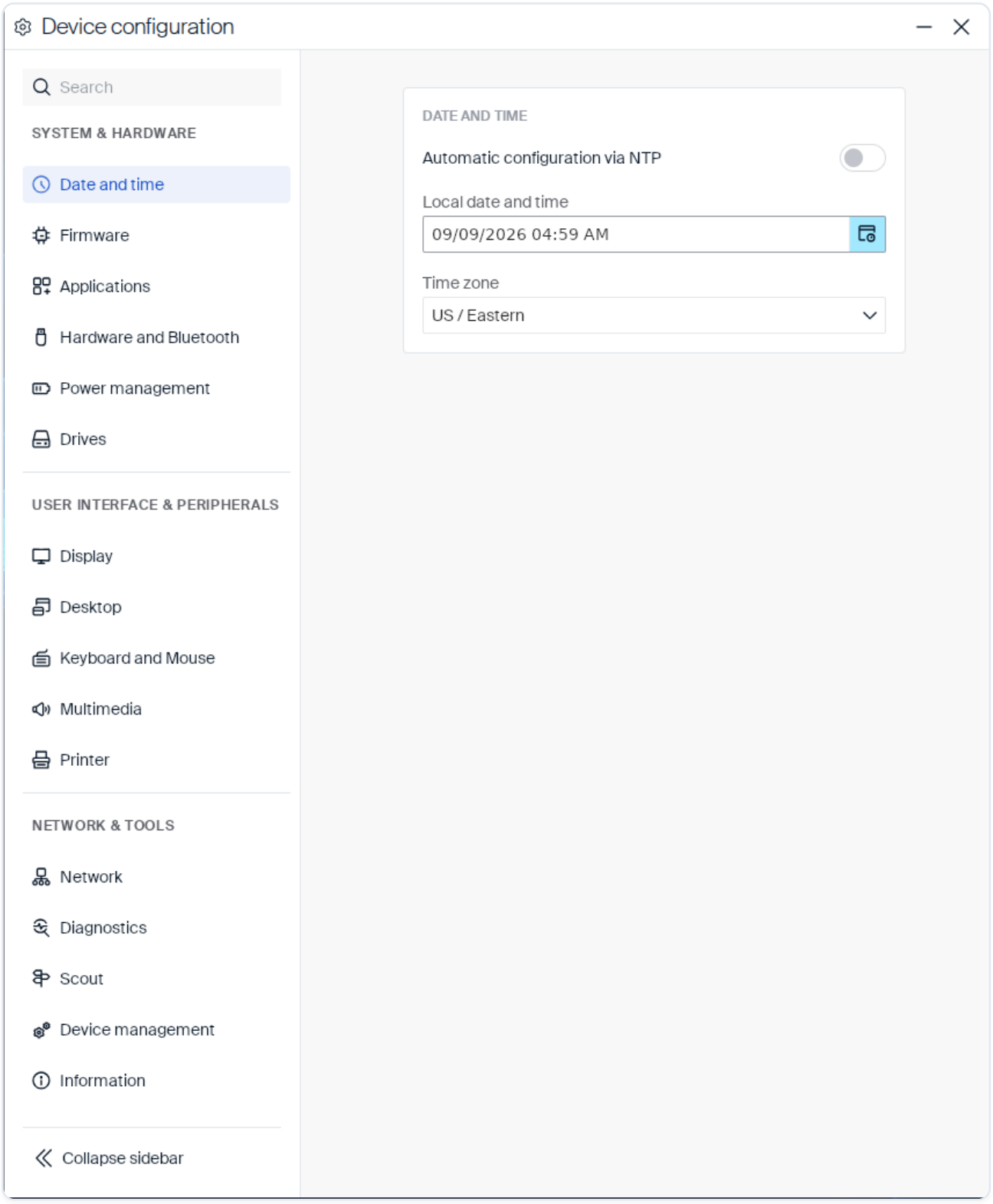
Note:

To persist a remembered device across reboots (when not covered by the whitelist above), add `audio` (or `ALL`) to `Global.PersistentHomeApplications` in Scout.

This only controls which device becomes the **default** — non-default devices remain usable and can still be selected manually. Applies to both audio output and input.

Date and time

Date and time is where you set the device's date and time, its time zone, and NTP (Network Time Protocol) synchronization.



Setting the date and time manually

1. Clear **Automatic configuration via NTP**.

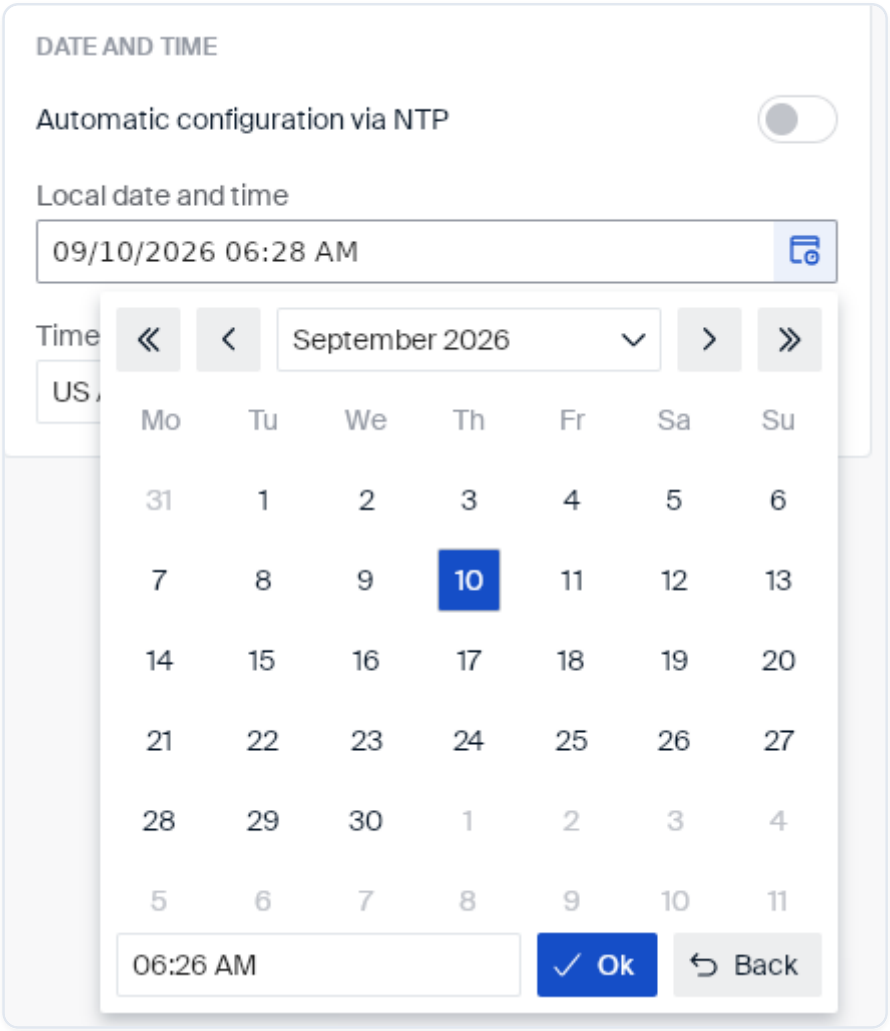
   The **Local date and time** field appears, in `MM/DD/YYYY hh:mm aa` format.

2. Enter the date and time directly, or click the picker button next to the field to open the date and time picker.

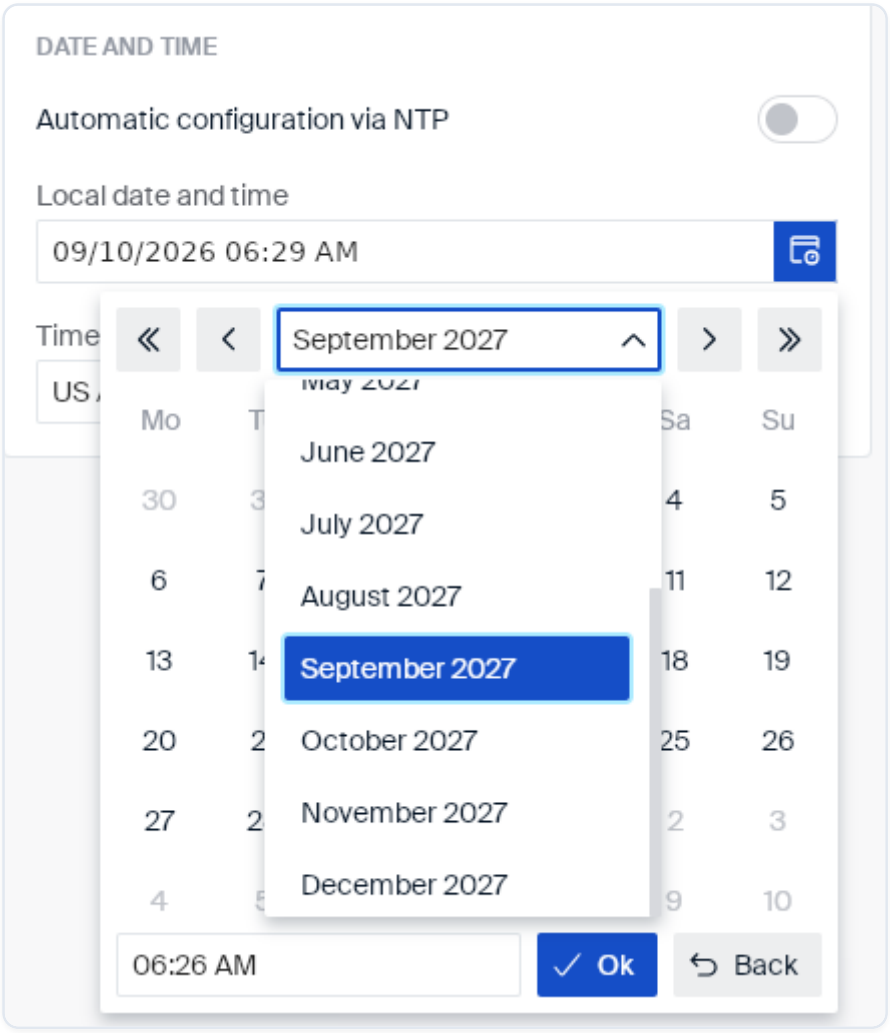
The change takes effect immediately.

USING THE DATE AND TIME PICKER

The picker shows a calendar for the current month, with the currently set date highlighted, and a time field below the calendar.



- Click a day in the calendar to select that date.
- Click **<** or **>** to move to the previous or next month.
- Click **<<** or **>>** to move to the previous or next year.
- Click the month and year label (for example, **September 2027**) to open a dropdown list of months. Scroll the list and click a month to jump directly to it.
- Click the time field to select and edit the hour, minute, or AM/PM segment.



Click **Ok** to apply the selected date and time to the **Local date and time** field, or click **Back** to close the picker without applying any change.

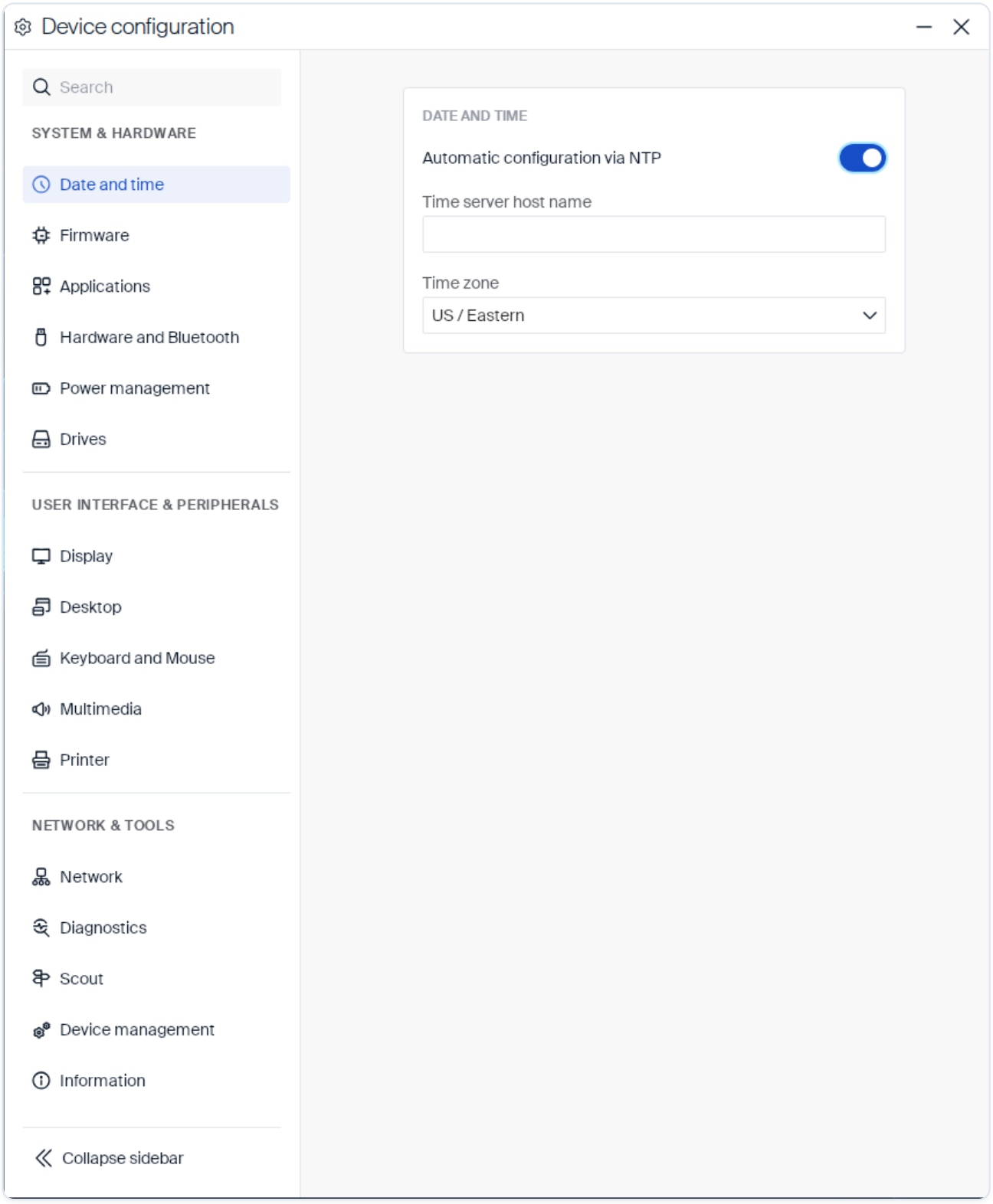
Synchronizing the date and time via NTP

1. Select **Automatic configuration via NTP**.

   The **Local date and time** field is replaced by a **Time server host name** field.

2. Optionally, enter the hostname of a specific NTP server to use.

The device keeps its clock synchronized automatically.

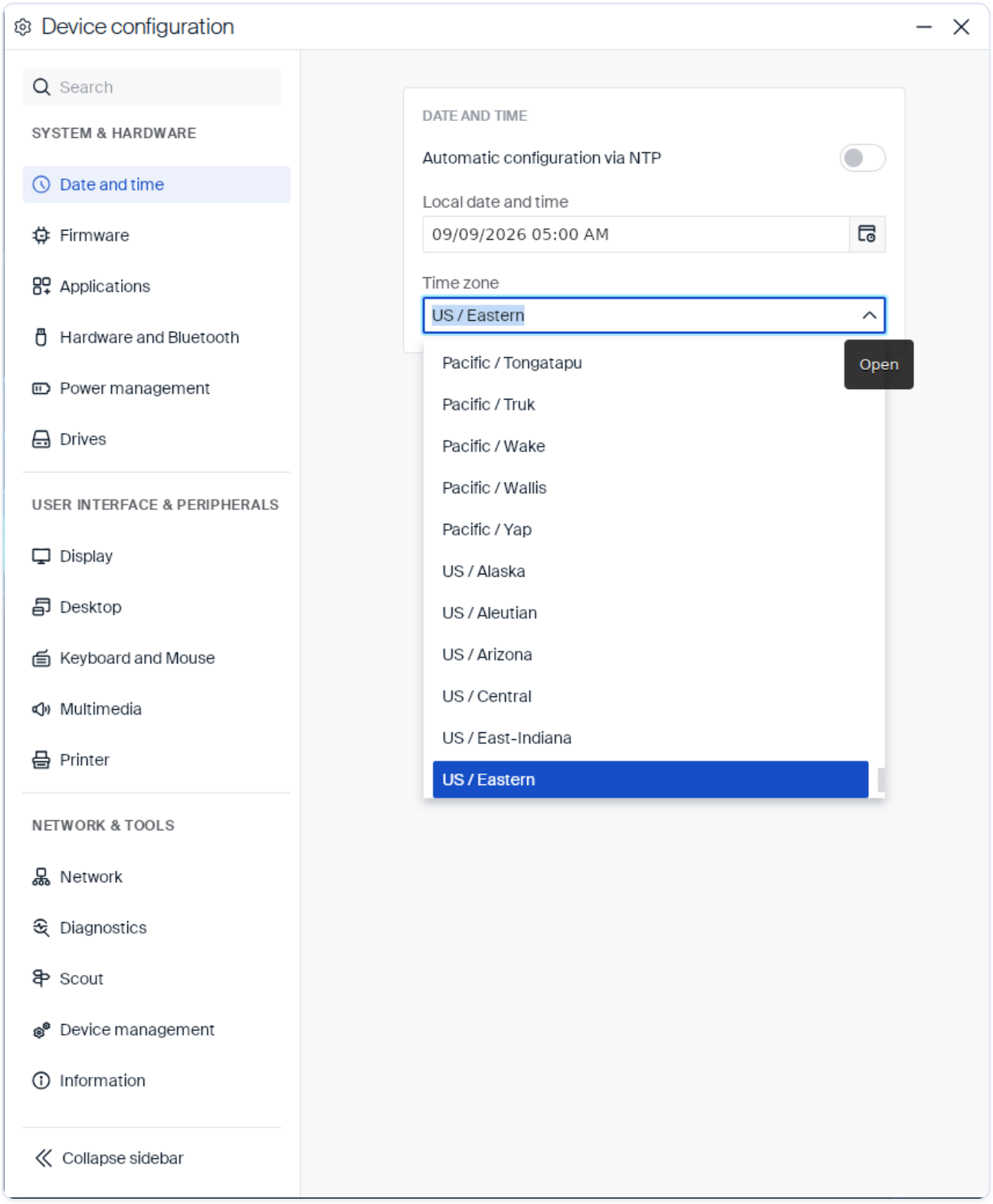


Setting the time zone

1. In the **Time zone** field, type to search for a region or city, or open the list to browse all available time zones.

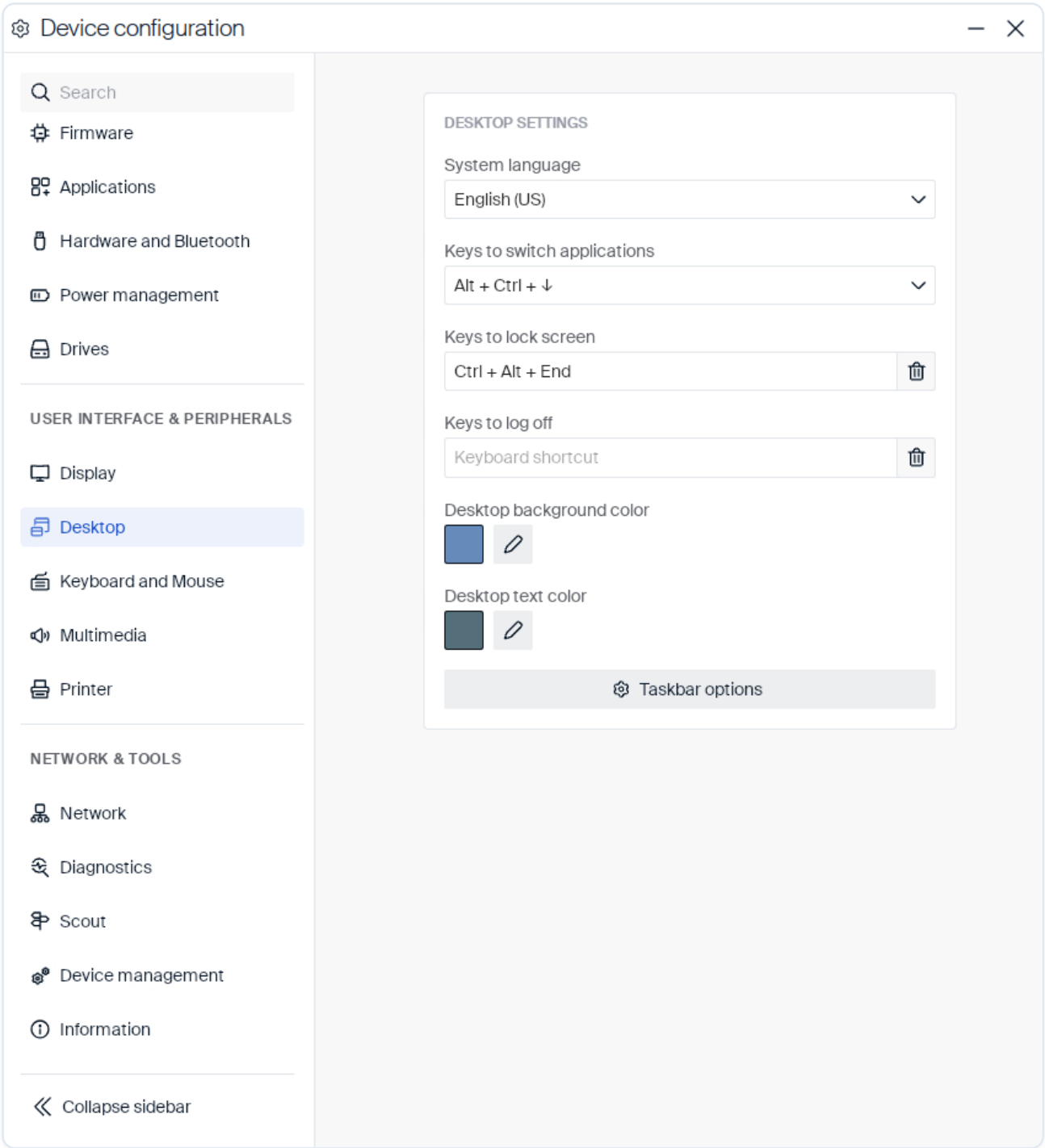
   Time zones are listed in `Region / City` format, for example `Europe / Berlin`.

2. Select the relevant time zone.



Desktop

Desktop is a section of Device configuration where you can set the system language, configure keyboard shortcuts for common actions, and customize the desktop's colors and system bar.



SETTING	TYPE	DESCRIPTION
System language	Dropdown	Sets the language of the user interface.
Keys to switch applications	Dropdown	Keyboard shortcut for switching between open applications: **Alt + Ctrl +** ↓ (default) or **Alt + Tab**.

SETTING	TYPE	DESCRIPTION
Keys to lock screen	Shortcut field	Keyboard shortcut for locking the screen. **Ctrl + Alt + End** by default.
Keys to log off	Shortcut field	Keyboard shortcut for logging off. Empty by default.
Desktop background color	Color picker	The color of the desktop background.
Desktop text color	Color picker	The color of desktop text and icon labels.

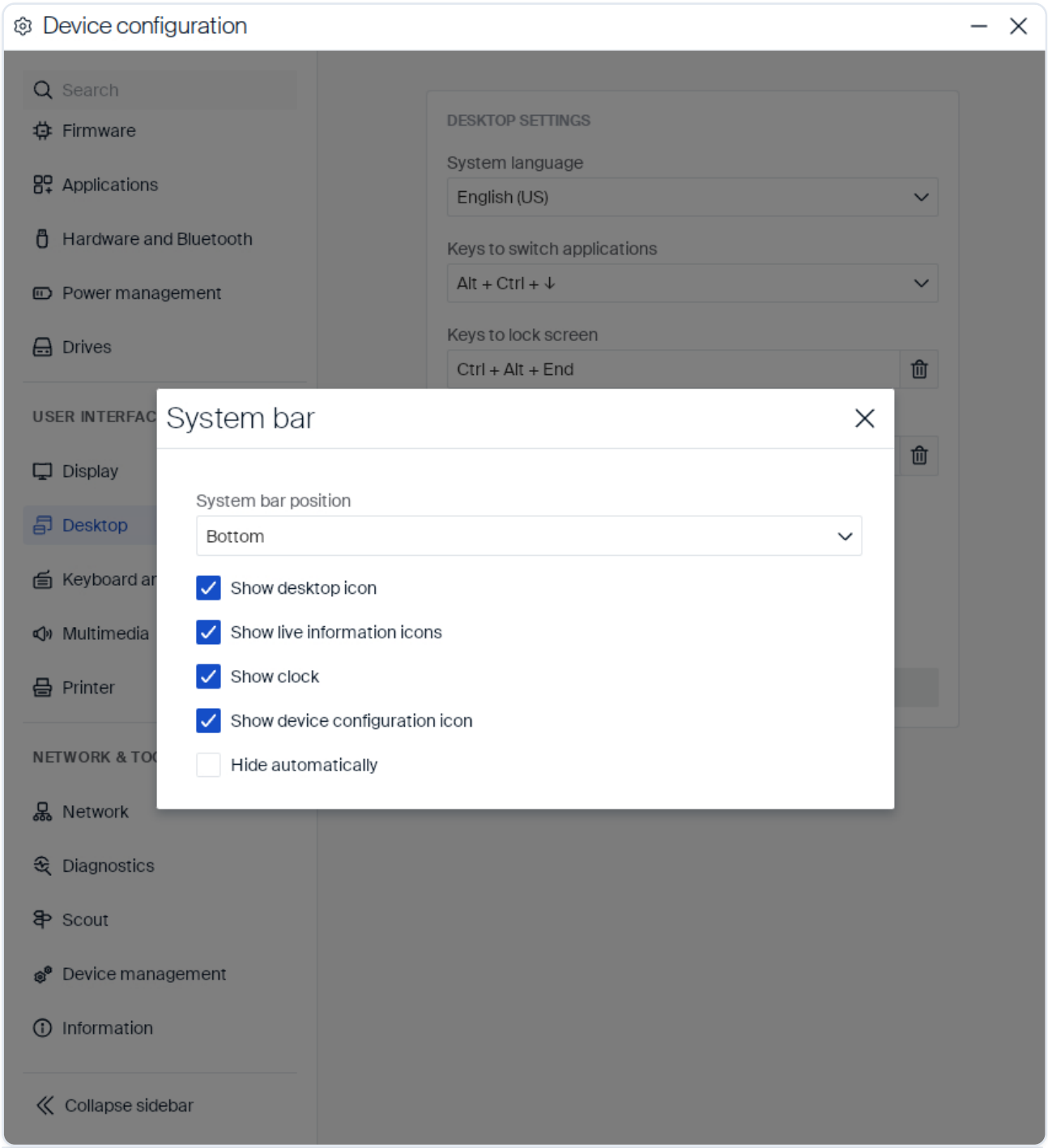
> **Note:**

> Click the trash icon next to a shortcut field to clear it. Click the pencil icon next to a color to open a color picker, where you can pick a color visually, drag the hue slider, or enter a HEX value.

> Selecting **Hebrew** as the system language switches the interface to a right-to-left (RTL) layout. Menus, dialogs, and the taskbar are mirrored accordingly.

System bar

Click **Taskbar options** to configure the system bar:



SETTING	TYPE	DESCRIPTION
System bar position	Dropdown	Top or bottom of the screen.
Show desktop icon	Checkbox	Shows the icon for minimizing all open windows.
Show live information icons	Checkbox	Shows live status icons, such as network and audio.
Show clock	Checkbox	Shows the current time.

SETTING	TYPE	DESCRIPTION
Show device configuration icon	Checkbox	Shows the icon for opening Device configuration.
Hide automatically	Checkbox	Hides the system bar when not in use. To show it, move the mouse pointer to the screen edge where the system bar is placed.

USING MORE THAN ONE MONITOR

With **Hide automatically** disabled, the system bar stays on the primary monitor.

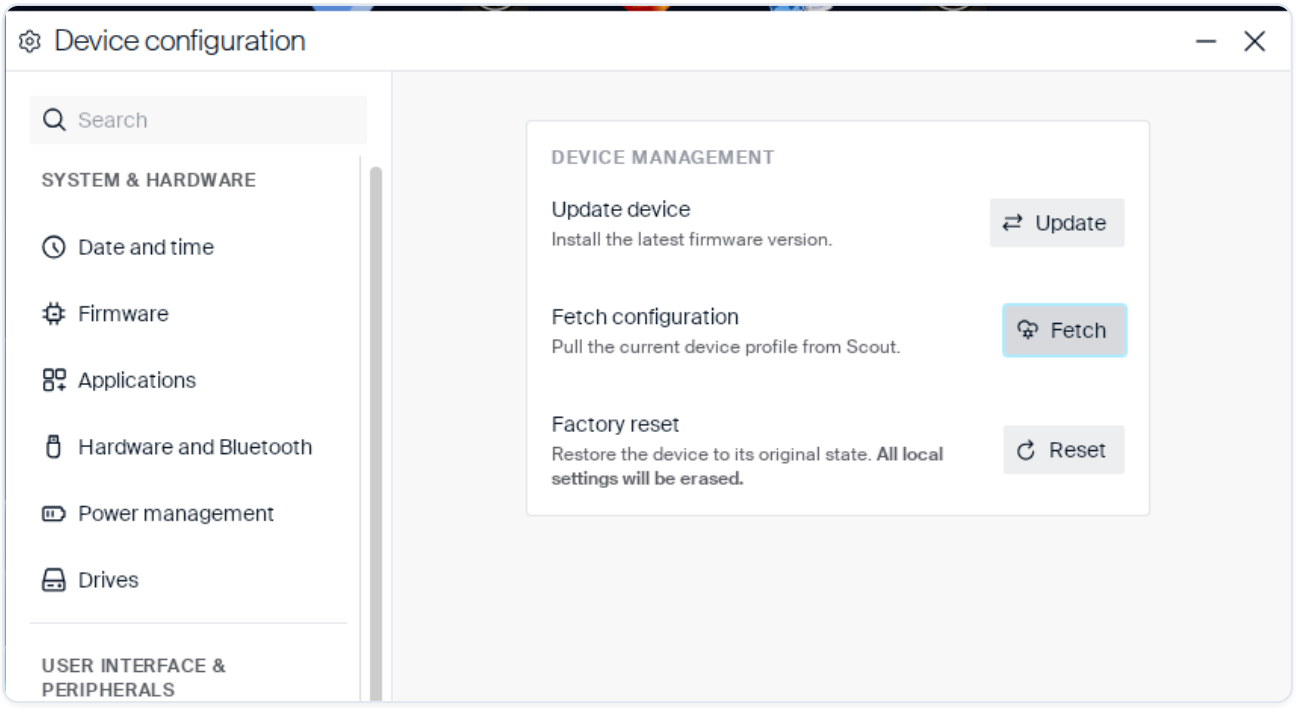
With **Hide automatically** enabled, the system bar appears on the monitor you are currently using, so you do not have to move the mouse to the primary monitor to reach it:

- To show the system bar, move the mouse pointer to the top or bottom edge of any monitor, depending on the **System bar position**.
- The first time the system bar is shown, it appears on the monitor where the mouse pointer is.
- The Start menu, Device configuration, and anything else you open from the system bar appear on the same monitor as the system bar.
- To close an open menu or panel, click an empty part of the desktop on any monitor.
- The system bar keeps working when you connect or disconnect a monitor, dock or undock a laptop, or change the screen resolution.

Device management

Device management is located under **Network & Tools** in the Device configuration dialog.

It provides firmware updates, configuration fetches, credential changes, and factory resets.



Each row is a single-click action:

ACTION	BUTTON	DESCRIPTION
Update device	**Update**	Checks for and installs the latest firmware version. Runs immediately (no confirmation prompt) — shows a **Firmware update** dialog with a progress indicator while it checks, then reports the result (for example **No update required.**).
Fetch configuration	**Fetch**	Pulls the current device profile from Scout. Shows a confirmation prompt first: **The device configuration will be updated. Continue?** with **Yes**/**Cancel**.
Change credentials	**Change**	Changes the device's authentication credentials (AD or Imprivata devices only).

ACTION	BUTTON	DESCRIPTION
Factory reset	**Reset**	Restores the device to its original state. **All local settings are erased.**

> **Note:**
>
> Which actions are visible depends on the user rights an administrator has defined in Scout.

> **Change credentials** appears only on AD or Imprivata devices — other devices show just **Update device**, **Fetch configuration**, and **Factory reset**.

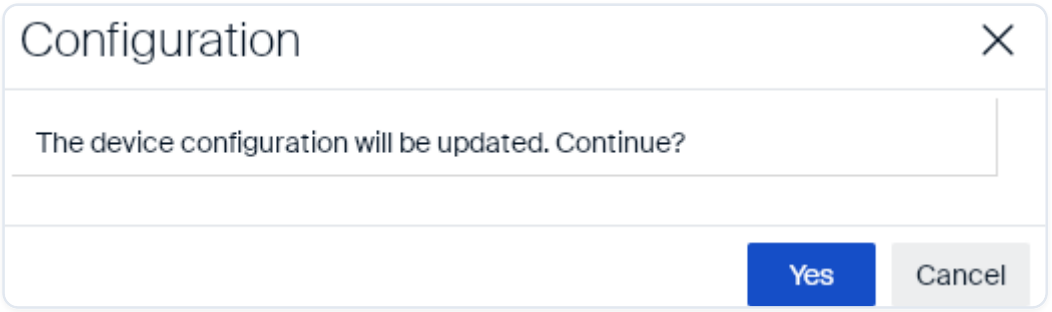
Update device

Before you click **Update**, check that the firmware settings are configured correctly. See Firmware.

After the software is installed, the device restarts.

Fetch configuration

Use **Fetch** to discard local changes to the device configuration or the application definitions and return to the settings that Scout defines. A prompt asks you to confirm:

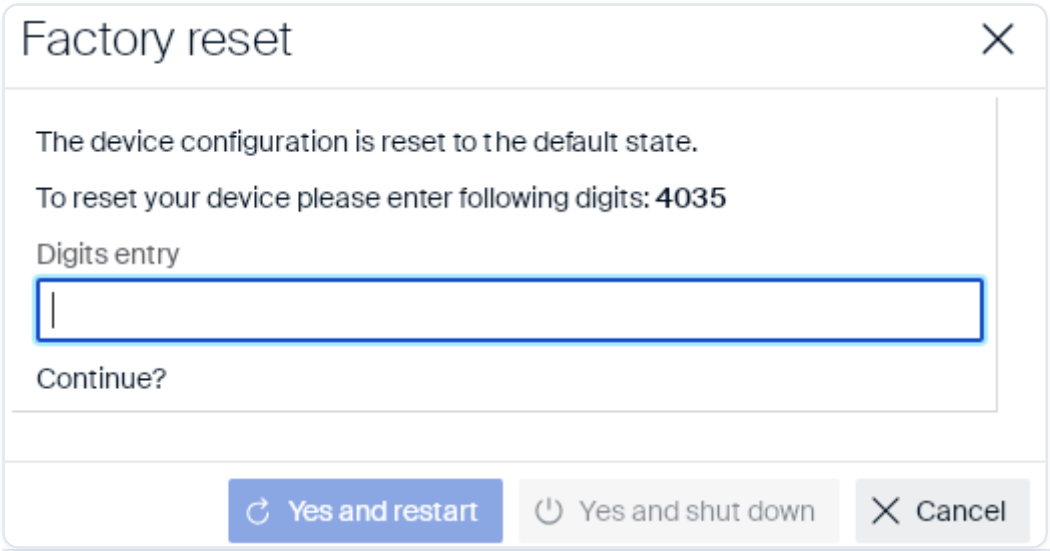


After you confirm, the current device configuration and application definitions for the device or its OU are loaded from the Scout server. Local settings are overridden, unless they are protected.

Factory reset

Use a factory reset for troubleshooting, for example when the local device configuration does not work correctly.

Clicking **Reset** opens a **Factory reset** dialog. As a safeguard against accidental resets, it displays a random 4-digit code that you must retype into **Digits entry** before continuing:



BUTTON	DESCRIPTION
Yes and restart	Resets the device, then restarts it.
Yes and shut down	Resets the device, then shuts it down.
Cancel	Closes the dialog without resetting anything.

A factory reset deletes the local device configuration, the local application definitions and the connection data to the Scout server. The installed firmware stays.

After the reset, the device starts as in its initial state. To connect it to Scout again, use one of these methods:

- The wizard on the device: First configuration.
- The DNS alias `ScoutSrv`.
- The DHCP options `222` and `223`.

The device then receives the configuration of the OU that it is assigned to. For details on DNS and DHCP, see Registering Scout Server in the network.

Note:

When an administrator resets a device remotely from Scout, the server address and the OU ID can be kept. The device then stays connected to Scout.

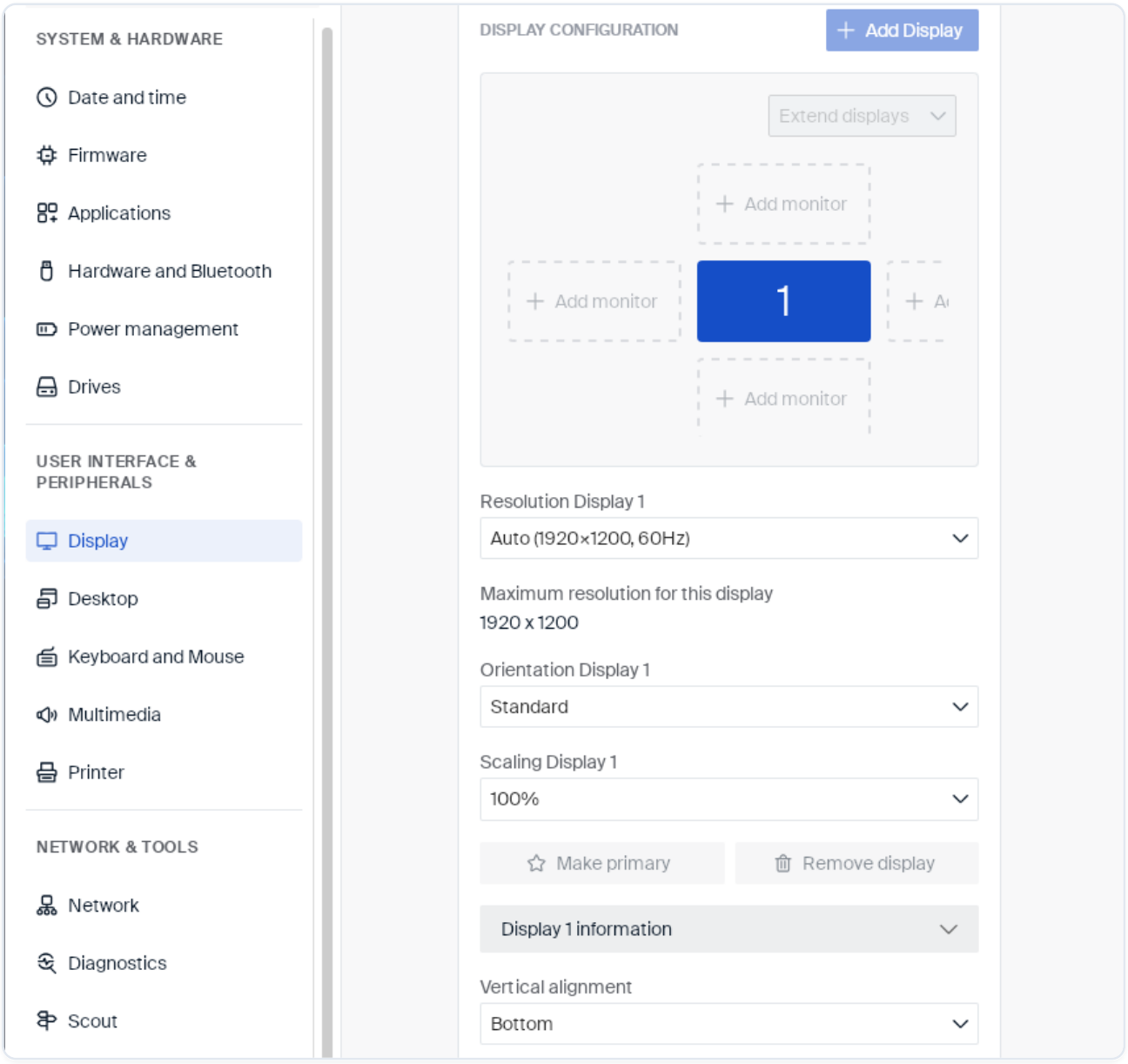
Scheduled commands

The device can also run recurring commands by itself. It can update the firmware, synchronize the device configuration, restart or shut down the device, and run user-defined commands. The device starts these commands at the times that you define, in its own time zone.

You define the commands in UniconOS Management Board, in the **UniconOS Command Scheduler** section of the advanced device configuration of the base configuration or an OU.

Display

Display is where you configure the monitor layout and screen resolution.



Arranging monitors

Connected monitors are shown as numbered boxes in the layout diagram:

1. Click **Add monitor** on the side where an additional monitor is physically positioned, to add it to the layout.
2. Click **Add Display** to add a monitor that isn't currently detected.
3. Use **Extend displays** to control whether connected monitors extend the desktop or mirror it.

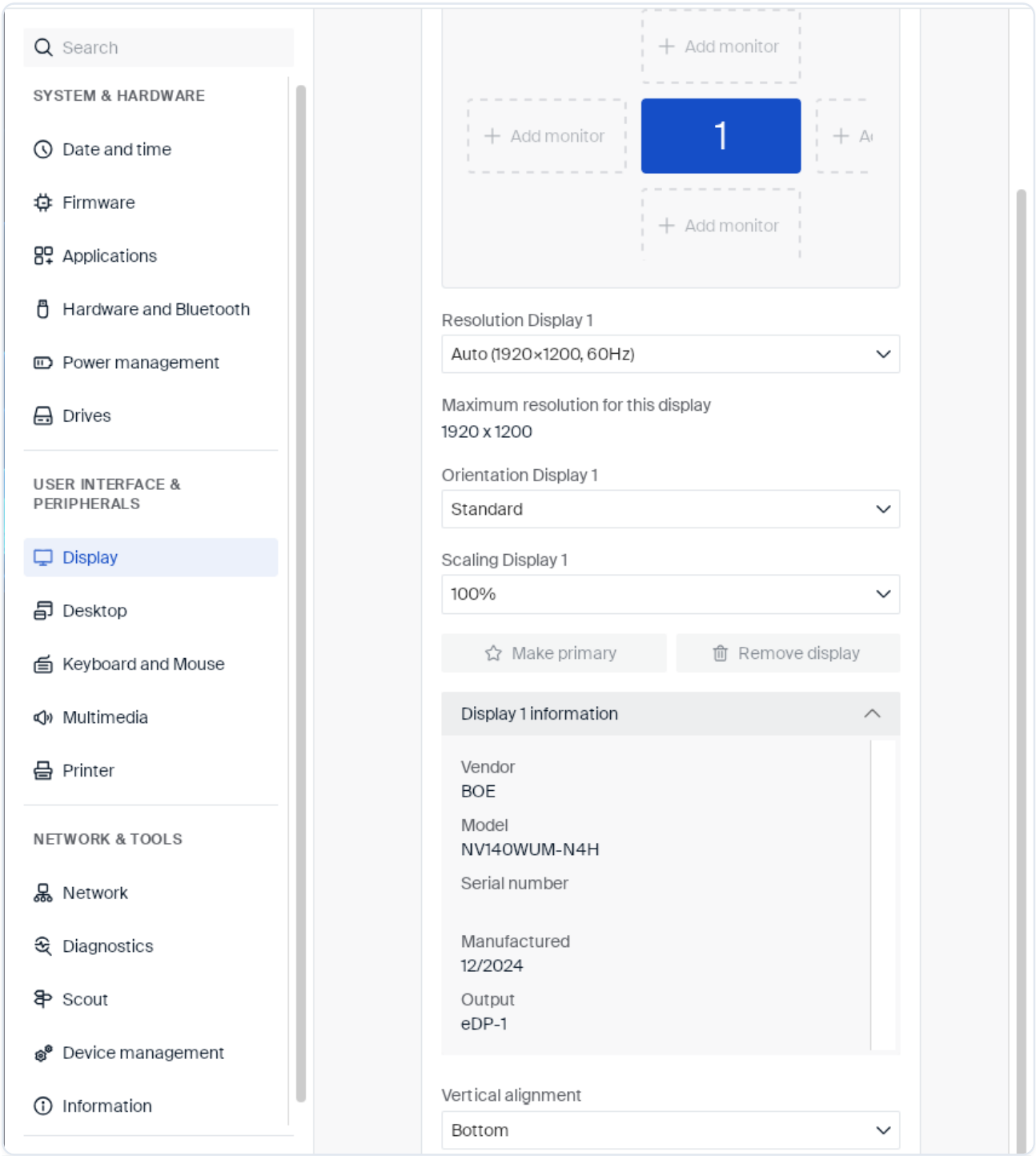
Per-display settings

Each connected display has its own settings, covering its resolution, rotation, and how it lines up with other displays:

SETTING	TYPE	DESCRIPTION
Resolution	Dropdown	The display's resolution and refresh rate, or `Auto` to use the maximum supported resolution. Below it, a **Maximum resolution for this display** line shows the highest resolution the display supports.
Orientation	Dropdown	Rotates the display's output.
Scaling	Dropdown	Sets the display's scaling percentage.
Vertical alignment	Dropdown	How the display lines up with other displays that have a different height, for example `Bottom`.

Click **Make primary** to set a display as the primary display, or **Remove display** to remove it from the layout. With only one display connected, both buttons are disabled.

Click **Display information** to expand further details about the selected display:



FIELD	DESCRIPTION
Vendor	The display panel's vendor.
Model	The display panel's model name.
Serial number	The display's serial number, if reported. Shown blank if not.
Manufactured	The display's manufacture date (month/year).

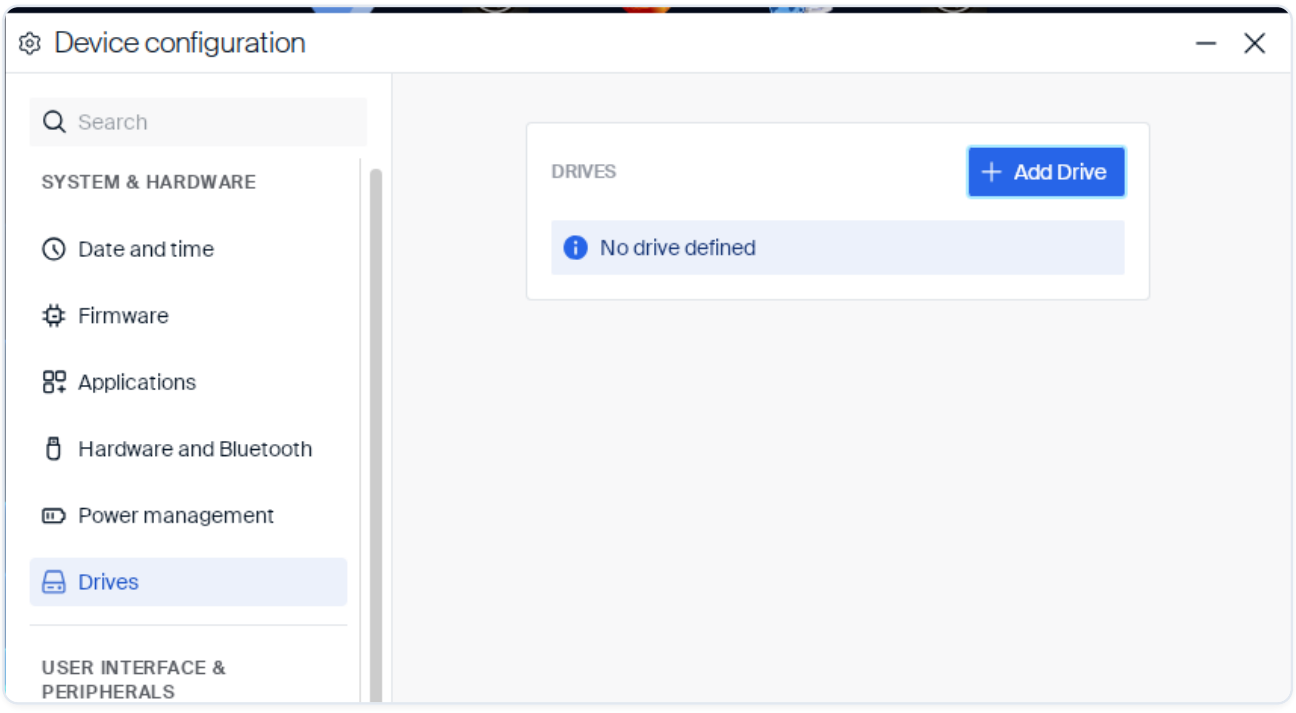
FIELD	DESCRIPTION
Output	The video output the display is connected to, for example `eDP-1` for an internal laptop panel.

> **Note:**
>
> While the Display page is open, each connected monitor shows a large identifying number overlay on screen (for example, a monitor labeled **1** shows a large **1**), matching the number in the layout diagram.

> **Note:**
>
> An administrator can lock individual display settings on this page through UniconOS Management Board User rights. A locked setting stays visible but greyed out.

Drives

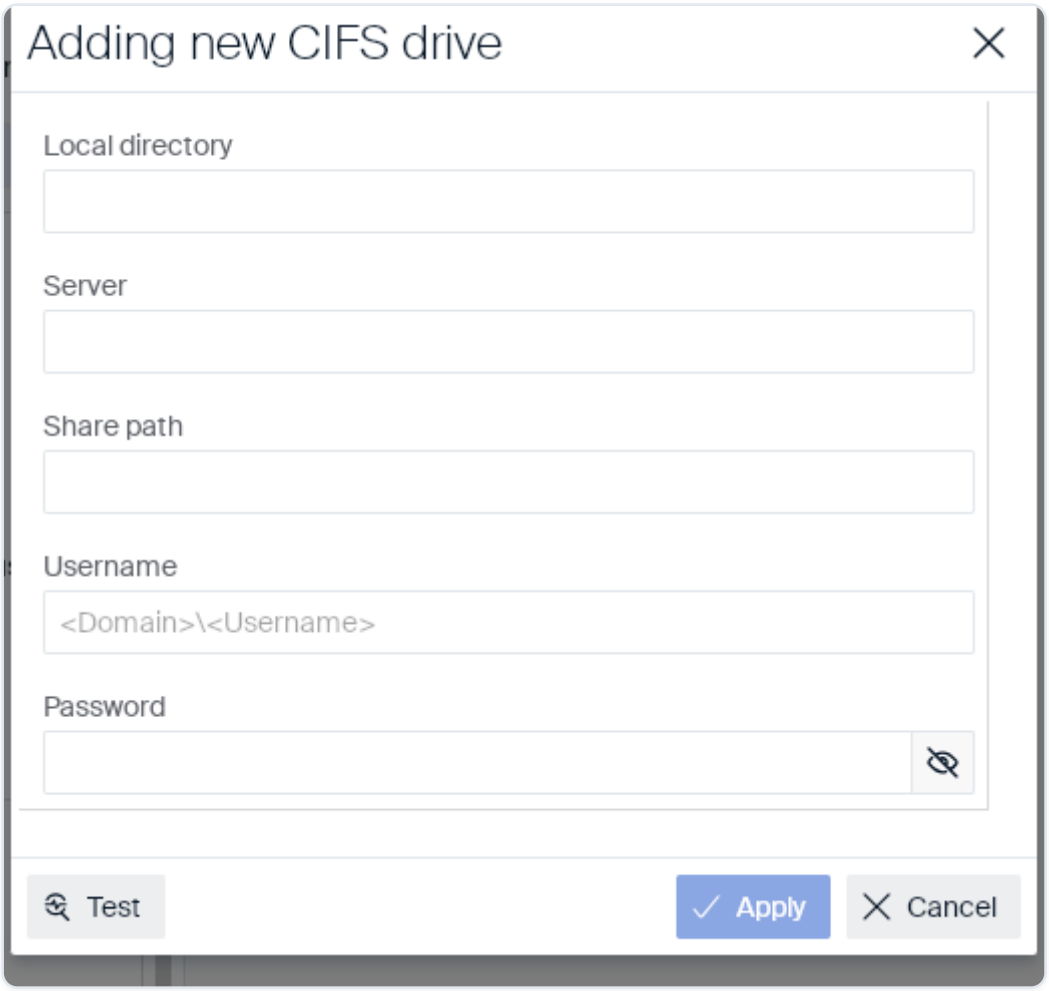
Drives is where you configure CIFS drives for accessing files and directories on the network.



When no drive is configured yet, the page shows a **No drive defined** notice next to **Add Drive**.

Adding a CIFS drive

1. Click **Add Drive**.

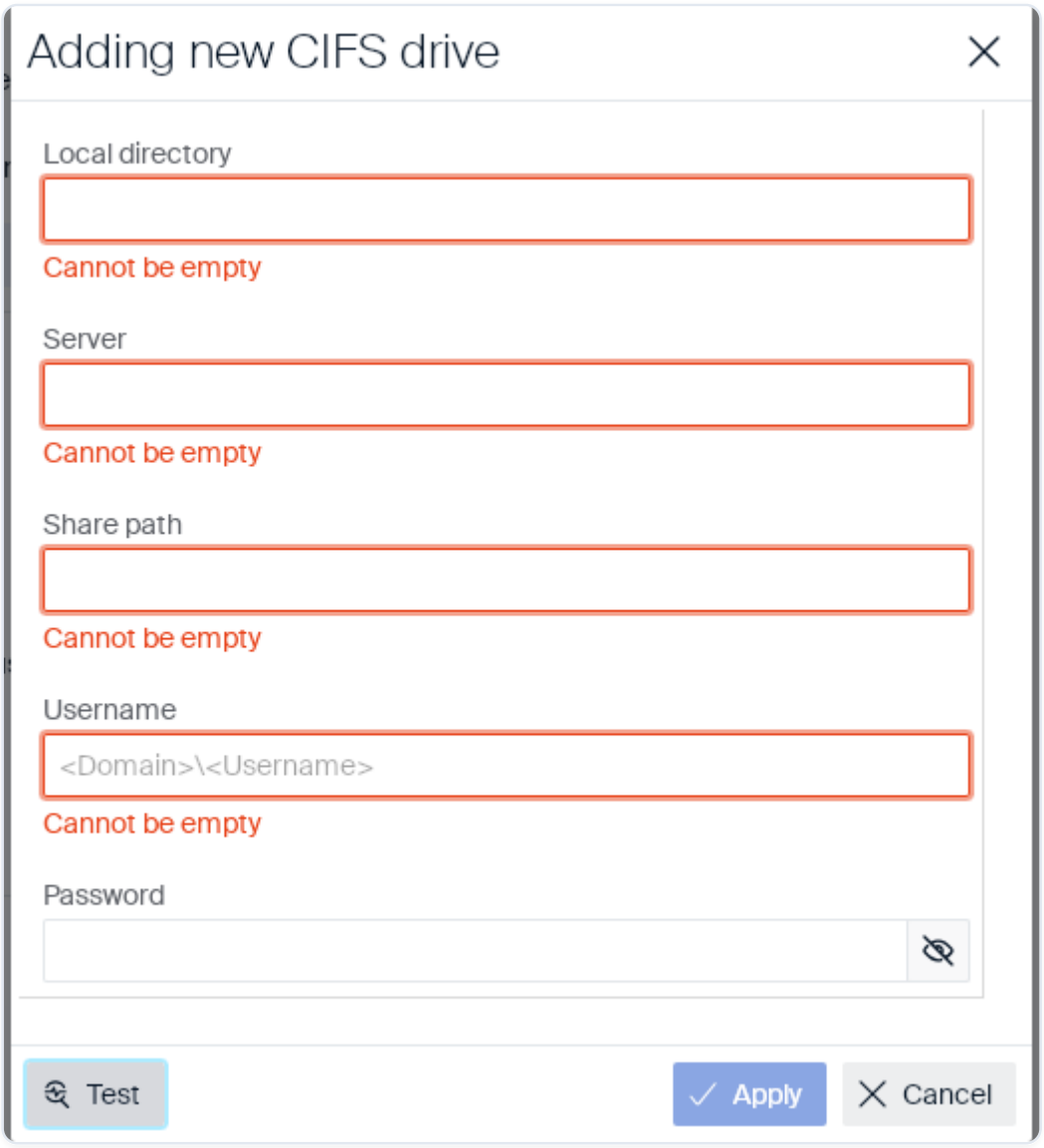


2. Fill in the following fields to identify the network share and the credentials used to connect to it:

FIELD	TYPE	DESCRIPTION
Local directory	Text field	Local mount point where the network share appears on the device. Required.
Server	Text field	Name or address of the CIFS/SMB server. Required.
Share path	Text field	Path of the share on the server. Required.
Username	Text field	User name for the share, in `<Domain>\<Username>` format. Required.

FIELD	TYPE	DESCRIPTION
Password	Text field	Password for the share. Has a show/hide (eye) icon. Not marked required.

Local directory, **Server**, **Share path**, and **Username** are mandatory — clicking **Test** or **Apply** while any of them is empty outlines that field in red with a **Cannot be empty** message and does not proceed:



3. Optionally, click **Test** to verify the connection before saving.
4. Click **Apply**.

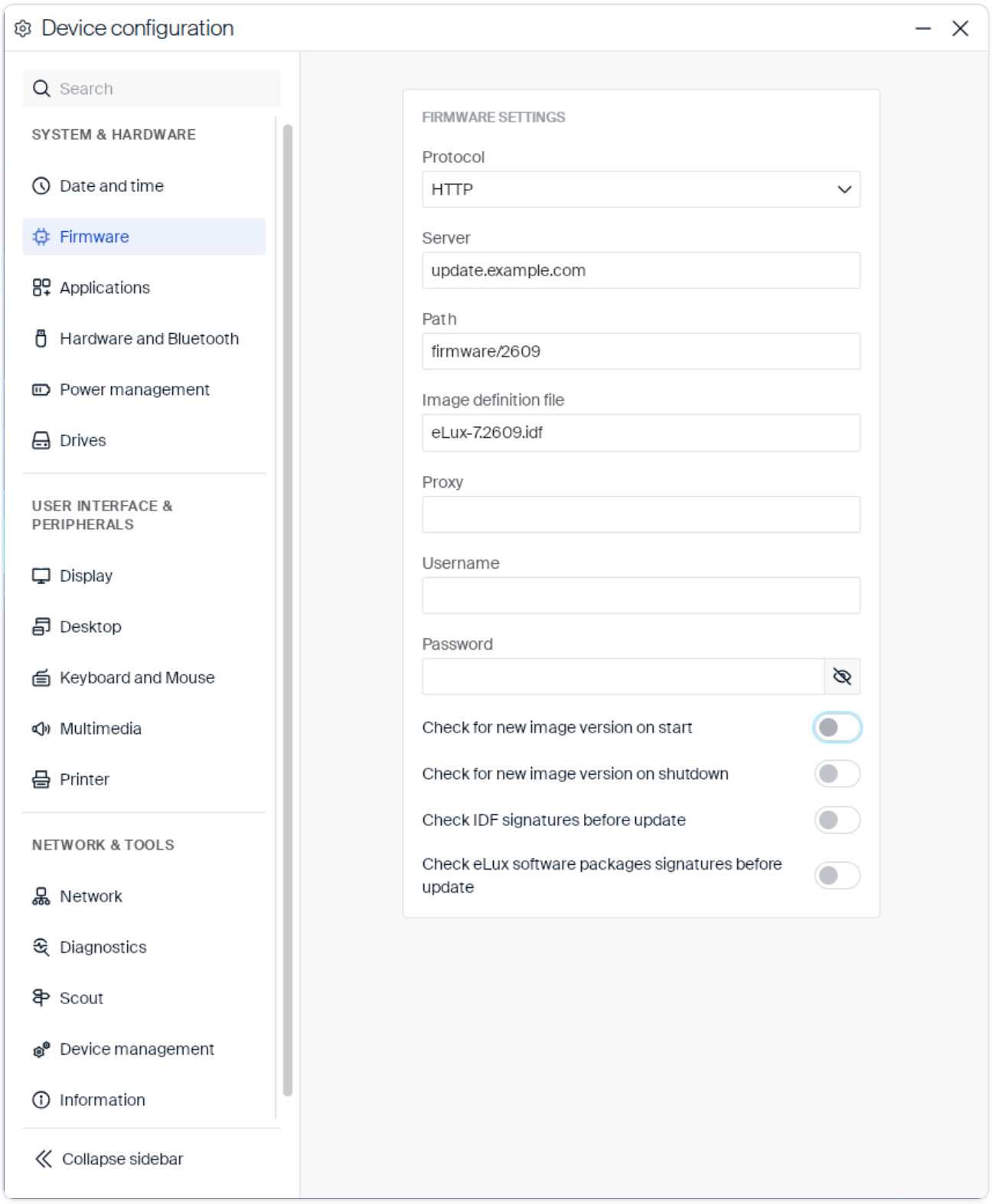
The drive appears in the drives list. To remove a drive, use its delete action in the list.

Firmware

Firmware is where you configure the image file and the firmware update procedure.

Firmware settings

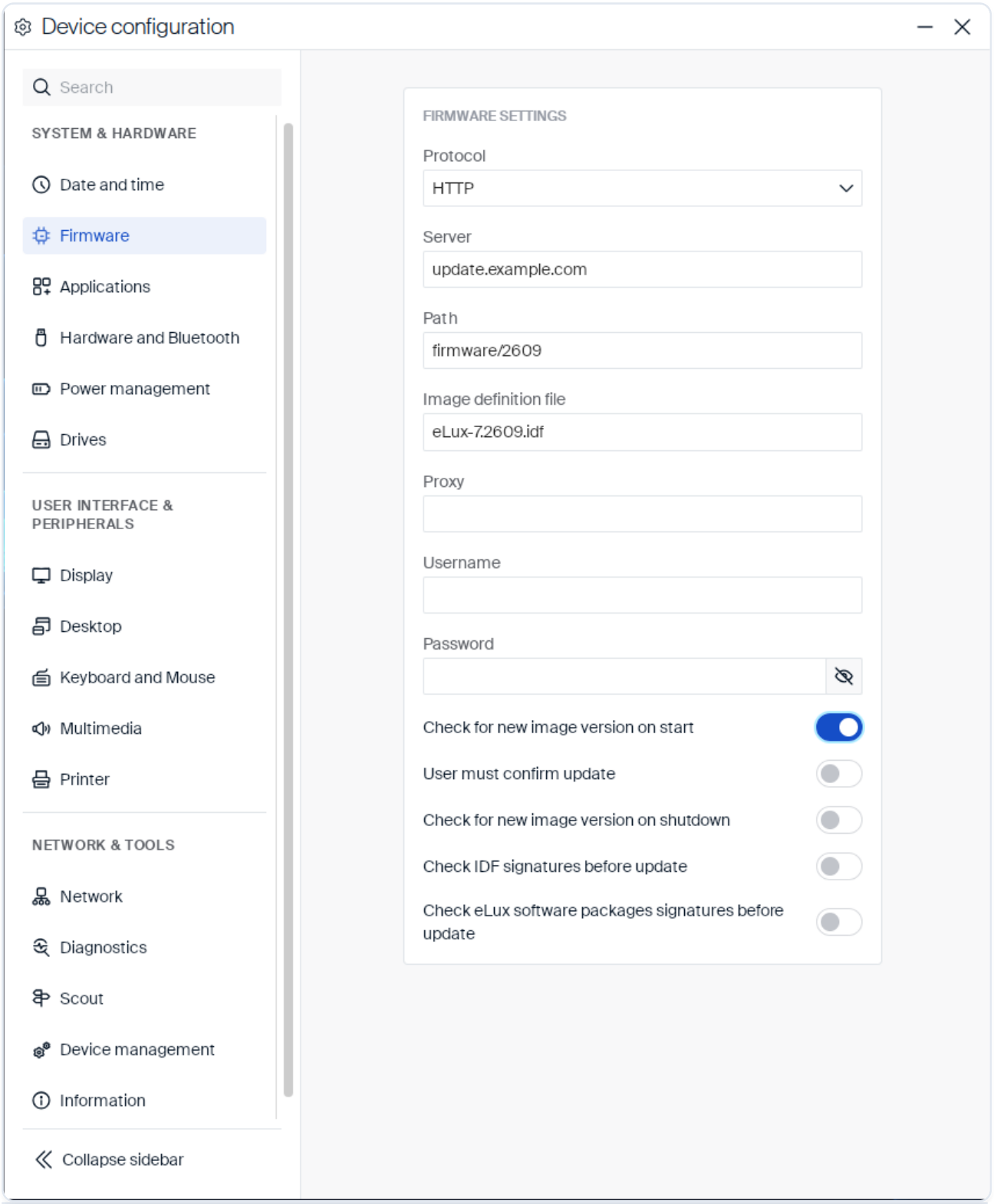
The Firmware section offers the following settings:



SETTING	TYPE	DESCRIPTION
Protocol	Dropdown	`HTTP`, `HTTPS`, `FTP`, or `FTPS`.
Server	Text field	Address of the update server.
Path	Text field	Directory on the server that contains the firmware packages and Image Definition Files.
Image definition file	Text field	Name of the Image Definition File (`.idf`) that describes the software packages to install.
Proxy	Text field	Proxy server to use for the update connection, if required.
Username	Text field	User name for the update server, if authentication is required.
Password	Text field	Password for the update server, if authentication is required. Click the eye icon next to the field to show or hide the entered password.
Check for new image version on start	Toggle	Checks for a new firmware version when the device starts.
Check for new image version on shutdown	Toggle	Checks for a new firmware version when the device shuts down.
Check IDF signatures before update	Toggle	Verifies the signature of the Image Definition File before installing an update.

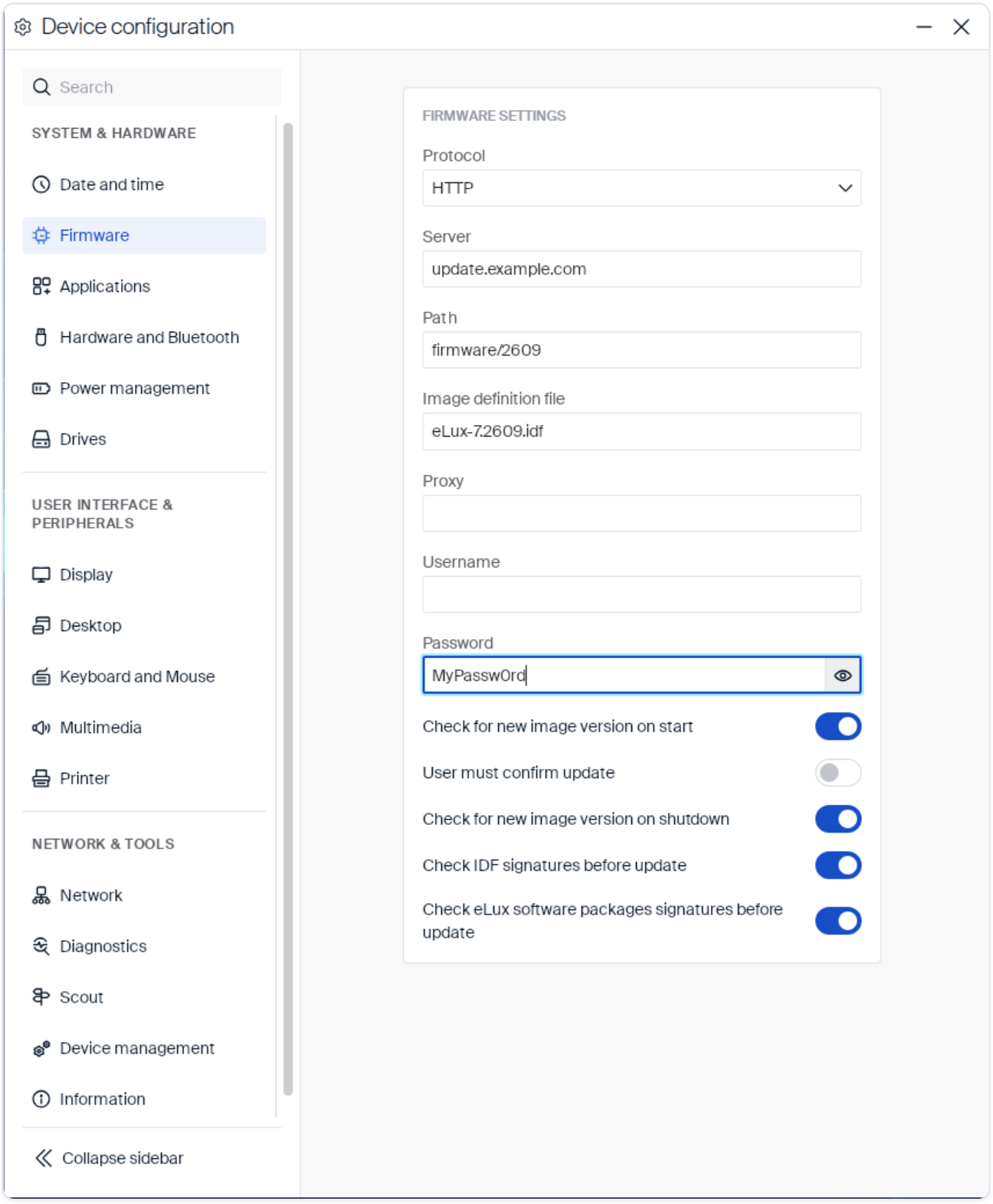
SETTING	TYPE	DESCRIPTION
Check eLux® software packages signatures before update	Toggle	Verifies the signature of the eLux software packages before installing an update.

Enabling **Check for new image version on start** reveals an additional **User must confirm update** toggle:



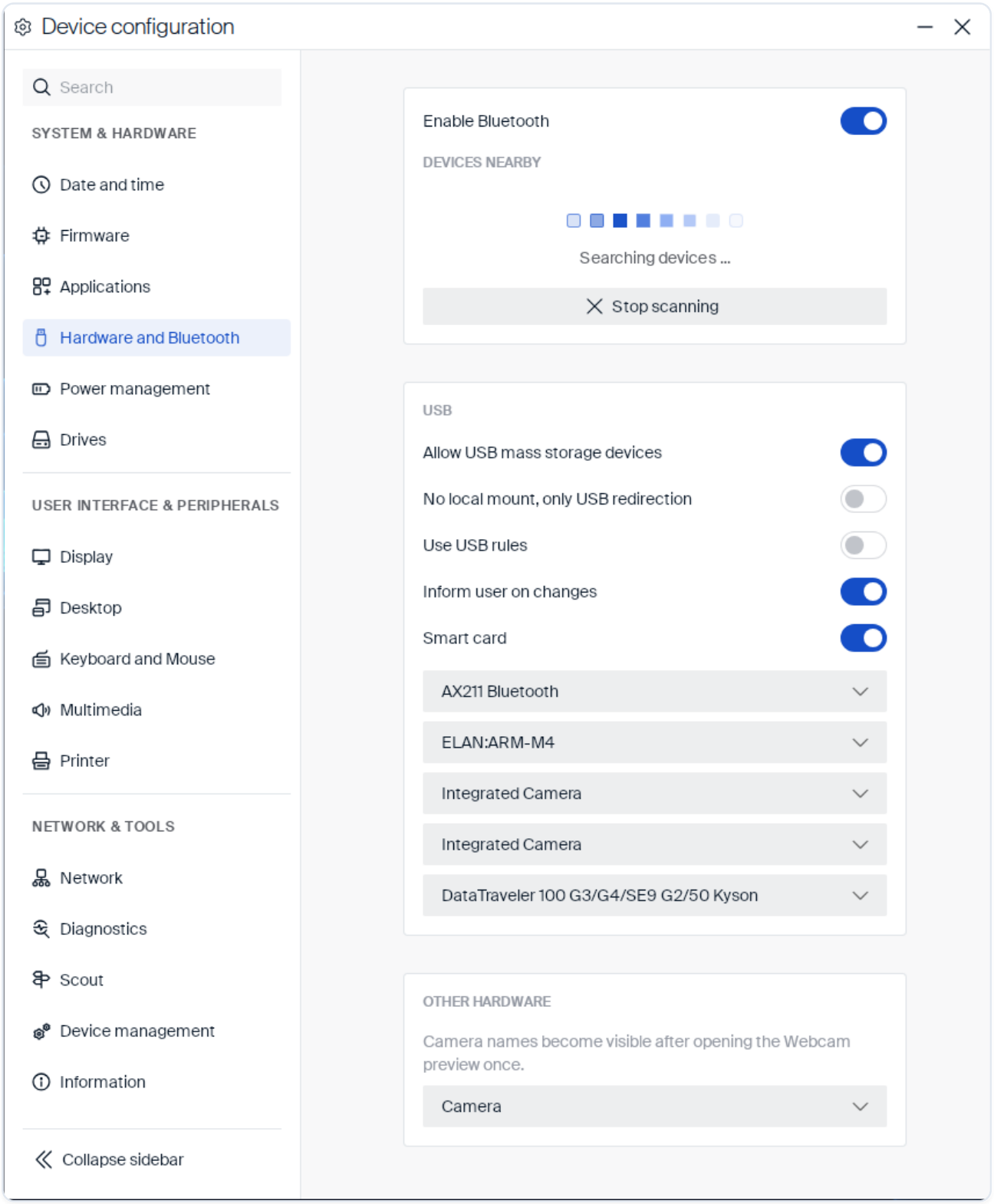
SETTING	TYPE	DESCRIPTION
User must confirm update	Toggle	Requires the user to confirm before an update is installed.

Clicking the eye icon next to the **Password** field shows the entered password as plain text:



Hardware and Bluetooth

Hardware and Bluetooth is where you configure the use of USB mass storage devices and smart cards, pair Bluetooth devices, and view detected hardware.



Bluetooth

1. Select **Enable Bluetooth**.

   A **DEVICES NEARBY** area appears, showing a **Start scanning** button.

2. Click **Start scanning** to search for nearby Bluetooth devices.

The button changes to **Stop scanning**, and a loading animation with the text **Searching devices ...** appears while the scan runs. Devices found nearby are listed here so you can pair them.

3. Click **Stop scanning** to end the search.

If **Enable Bluetooth** is already selected when you open **Hardware and Bluetooth**, the scan starts automatically.

Turning off **Enable Bluetooth** hides the **DEVICES NEARBY** area completely.

USB and smart card

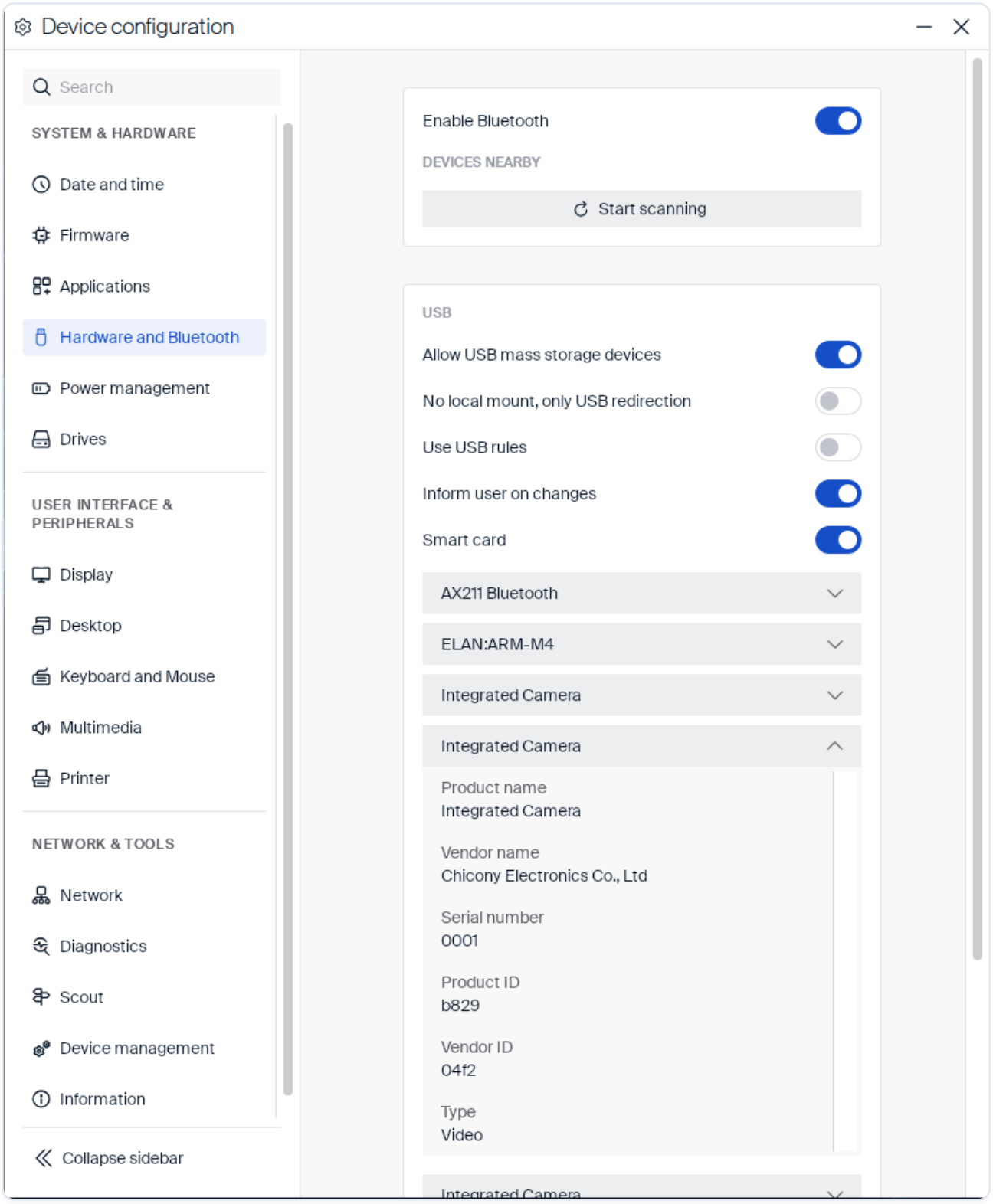
These toggles control whether USB storage devices and smart card readers can be used on the device, and how:

SETTING	TYPE	DESCRIPTION
Allow USB mass storage devices	Toggle	Allows USB storage devices (for example USB flash drives) to be used on the device.
No local mount, only USB redirection	Toggle	Prevents USB storage devices from being mounted locally; they're only available through USB redirection into a session. Only shown when **Allow USB mass storage devices** is on.
Use USB rules	Toggle	Applies administrator-defined USB rules instead of the default behavior.
Inform user on changes	Toggle	Shows a notification when a USB device is connected or disconnected.
Smart card	Toggle	Allows smart card readers to be used for certificate-based logon.

From UniconOS 2611 on, the device detects the smart card reader built into the Lenovo USB Wired Smartcard Keyboard 5000 without a custom driver bundle.

Detected hardware is listed below these toggles, in the same area. This can include the built-in Bluetooth adapter, touchpad, and camera, as well as any USB storage device currently connected. Click a row to expand it and view its details: **Product name**, **Vendor name**, **Product ID**, **Vendor ID**, and **Type** (for example **Video** or **Generic**). Some rows, such as a camera, also show a **Serial number**.

A physical device that exposes more than one USB interface can appear as more than one row. For example, an integrated webcam can be listed up to three times. These rows share the same Product ID, Vendor ID, and serial number, but their Type can differ.



Other hardware

> **Note:**

> Camera names become visible only after opening the webcam preview once.

> Below this note, a **Camera** row represents the device's webcam preview. It keeps the generic name **Camera** until the webcam preview has been opened once elsewhere on the device. Expanding this row only shows **Type: Webcam preview**, since it isn't a physical detected-hardware entry like the rows listed under **USB and smart card**.

Keyboard and Mouse

Keyboard and Mouse is where you set the keyboard language and functions, and configure mouse and touchpad settings.

Keyboard settings

These settings control the keyboard layout and how it behaves while typing:

SETTING	TYPE	DESCRIPTION
Language	Dropdown	The keyboard layout, for example `English (US, International)`.
Num Lock	Dropdown	Whether Num Lock is `On`, `Off`, or `Auto` at startup.
Delay	Slider	How long a key must be held before it starts repeating.
Speed	Slider	How fast a held key repeats.
Dead keys	Toggle	Enables dead keys, for typing combined characters as a key combination, for example **ê** or **ô**.
Console switch	Toggle	Allows switching to a virtual console with a keyboard shortcut. Changing this setting prompts you to log off to apply it.
Extended keys	Toggle	Enables multimedia and other keys with special functions.

The **Language** list includes two Dvorak layouts: `English (US, Dvorak)` and `English (UK, Dvorak)`.

> **Note:**

> For the **Turkish** language, two keyboard layout variants are available in the **Language** dropdown: **Turkish (Q)**, the most common Turkish layout, and **Turkish (F)**. Choose the variant that matches the layout you already use.

> The on-screen keyboard has its own settings in the **Accessibility** section of the device configuration. There, for example, **Hide keyboard close button** removes the keyboard's close button, so that users can't close the keyboard by accident. In UniconOS Management Board, you configure the on-screen keyboard in the **Keyboard and Mouse** section of the device configuration, under **Virtual keyboard**. For details, see On-Screen Keyboard.

Mouse settings

These settings control the size and speed of the mouse pointer, and how the mouse buttons work:

SETTING	TYPE	DESCRIPTION
Pointer size	Slider	The pointer size, from **Normal** to **Largest**. A larger pointer is easier to see.
Double-click speed	Slider	How fast two clicks must follow each other to count as a double-click.
Acceleration	Slider	How much the pointer speeds up with faster mouse movement.
Left-handed	Toggle	Swaps the primary and secondary mouse buttons for left-handed use.

> **Note:**

> Earlier versions had a **Show mouse pointer** switch in the mouse settings. It is no longer on the device. Administrators can still hide the pointer, for example on an information board. In UniconOS Management Board, turn off **Show mouse pointer** under **Keyboard and Mouse** > **Mouse** in the device configuration, see Device configuration / Keyboard and Mouse.

Touchpad settings

These settings control the touchpad and how it scrolls. They apply only to devices with a built-in touchpad:

SETTING	TYPE	DESCRIPTION
Touchpad	Dropdown	Turns the touchpad `On` or `Off`, or sets it to `Auto`.
Classic scroll direction	Toggle	Sets which way the page moves when you scroll on the touchpad.

The **Classic scroll direction** toggle has two settings:

- **Off** (default): natural scrolling. The page moves the same way as your fingers, as on macOS.
- **On**: classic scrolling. The page moves the opposite way, as on Windows.

For the other touchpad gestures, such as zooming a web page, see Touchpad gestures.

Setting values centrally

Scout and UniconOS Management Board have no dedicated settings for the pointer size or the scroll direction. To set them for several devices at once, or to select a Dvorak layout, use advanced file entries. For how to define an entry, see Advanced file entries.

All three keys are in the `Screen` section of `/setup/terminal.ini`:

KEY	VALUES	DESCRIPTION
`PointerSize`	`24`, `32`, `48`, `64`, `96`	The pointer size in pixels. `24` is **Normal**. `96` is **Largest**.
`ClassicScrollDirectionForTouchpad`	`true`, `false`	`true`: classic scrolling. `false` (default): natural.

KEY	VALUES	DESCRIPTION
`XkbLayout`	`en_US_dvorak`, `gb_dvorak`	The keyboard layout. These two values select a Dvorak layout.

`en_US_dvorak` selects `English (US, Dvorak)`. `gb_dvorak` selects `English (UK, Dvorak)`.

Setting additional keyboard (XKB) options

Some keyboard layouts use a comma (,) as the decimal separator on the numeric keypad. To make it always produce a period (.) instead, or to apply other XKB options, add the option to the keyboard layout in the eLux® keyboard database.

> **Note:**
>
> Setting `XkbOptions=` under `[Screen]` in `terminal.ini` has no effect in eLux 7 and should not be used.

1. Edit the keyboard database file `/setup/elux/kbd.ini`.
2. In the section for the keyboard layout used on the device, set the `options=` value. The following example sets the numeric keypad decimal key to a period for the German layout (`[de]`):

   ```
   [de]

   options=kpdl:dot
   ```

   You can specify multiple options separated by commas, for example `options=kpdl:dot,compose:ralt`. Set the option on the section that matches the layout the device actually uses.
3. Deploy the modified `/setup/elux/kbd.ini` to the device through Scout (see Files configured for transfer), then restart the device so the file is in place at startup.

To confirm the option is active, run the following command on the device:

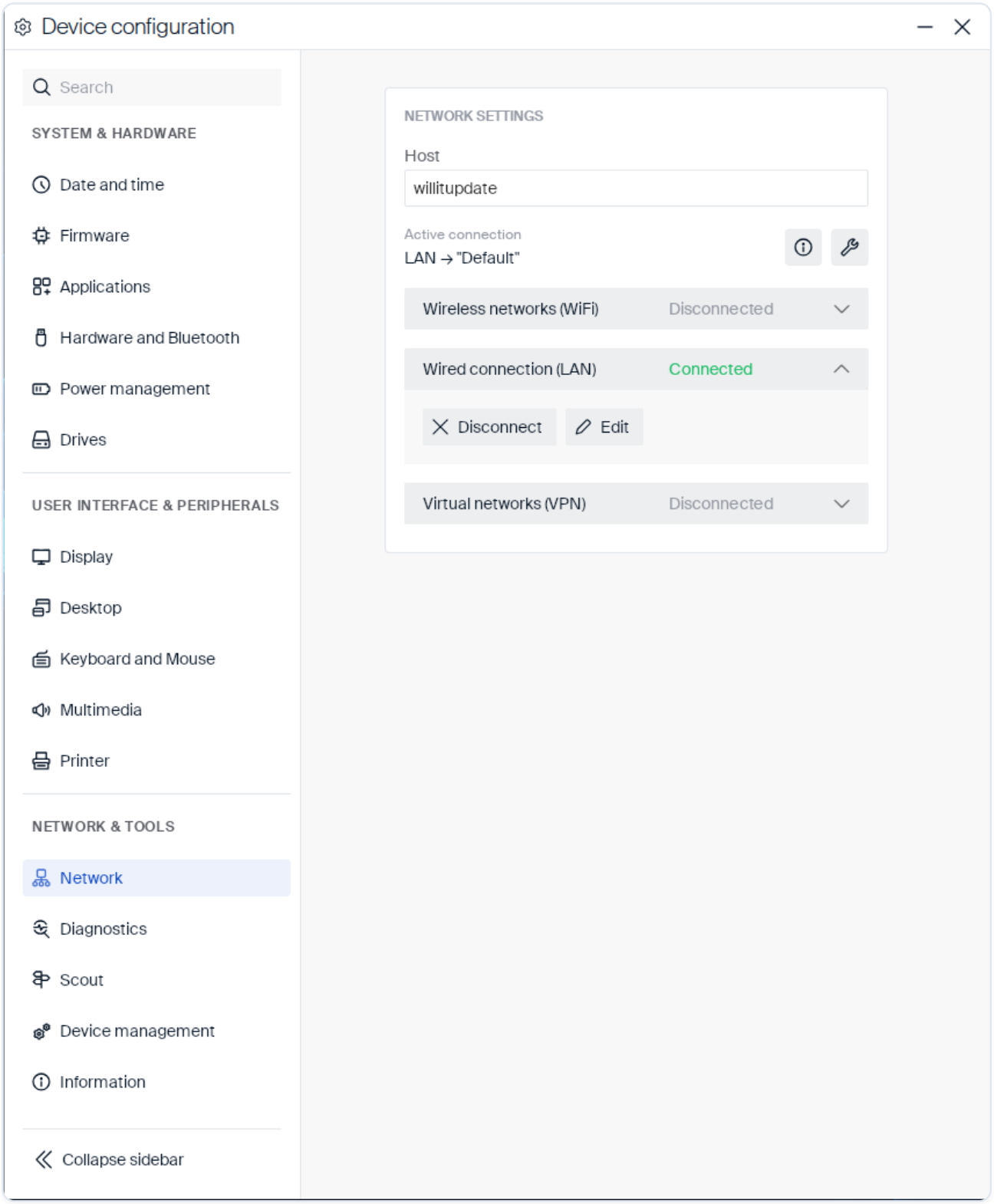
```
setxkbmap -query
```

The output lists the layout and its options, for example `options: kpdl:dot`.

> **Note:**
>
> If you later change the device's keyboard layout, add the option to the new layout's section in `kbd.ini` as well. An option set on one layout section does not apply to another.

Network

Network is where you configure the device's network interfaces and view information about the active connection. Available interfaces depend on the device's hardware and installed software packages.



The host name and the currently active connection are shown at the top of the page:

SETTING	TYPE	DESCRIPTION
Host	Text field	The device's host name.
Active connection	Info	The interface and profile that the device uses for its

SETTING	TYPE	DESCRIPTION
		network traffic, for example `LAN → "Default"`. If Ethernet and Wi-Fi are connected at the same time, this is the preferred connection. Click the info icon for connection details, or the wrench icon for advanced settings.

Below the active connection, each available interface type is listed with its status, for example **Connected** or **Disconnected**:

- **Wireless networks (WiFi)** — shows available wireless networks to connect to.
- **Wired connection (LAN)** — the device's Ethernet connection.
- **Virtual networks (VPN)** — VPN connections.
- **Mobile internet (WWAN)** — mobile broadband connections through a SIM card or an eSIM. Shown only if the device has a mobile broadband modem, see Mobile internet and eSIM.

Click an interface to expand it. For a connected interface, click **Disconnect** to disconnect it, or **Edit** to change its connection settings. When you open **Network**, the interface of the active connection is already expanded.

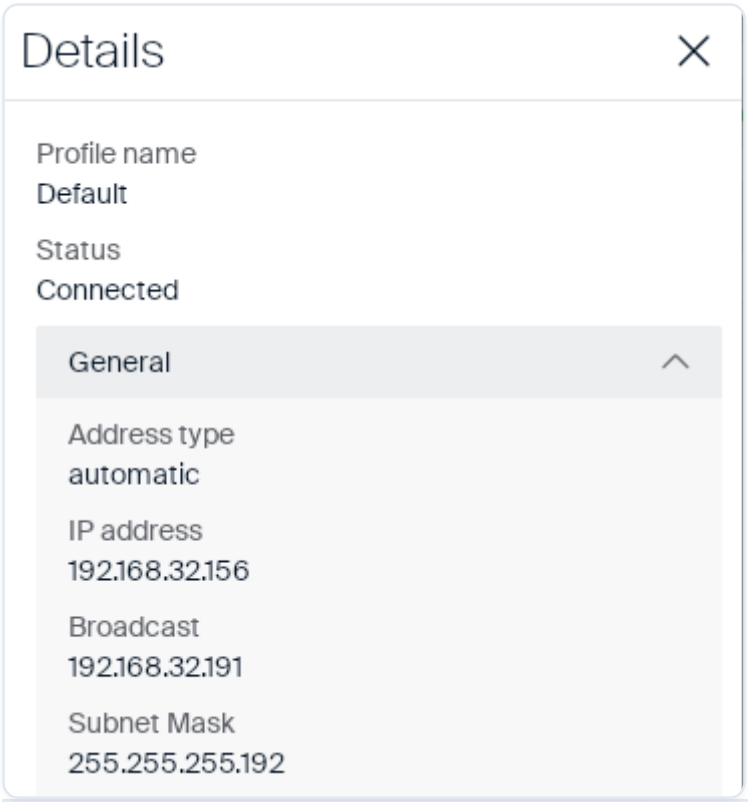
If an administrator allows parallel connections, Ethernet and Wi-Fi can be connected at the same time. See Using Ethernet and Wi-Fi at the same time.

Viewing connection details

To see the details of the active connection, click the info icon next to **Active connection**. The **Details** dialog shows the profile name and connection status, followed by three collapsible sections: **General**, **Advanced**, and **Statistics**. Click a section's header to expand or collapse it.

GENERAL

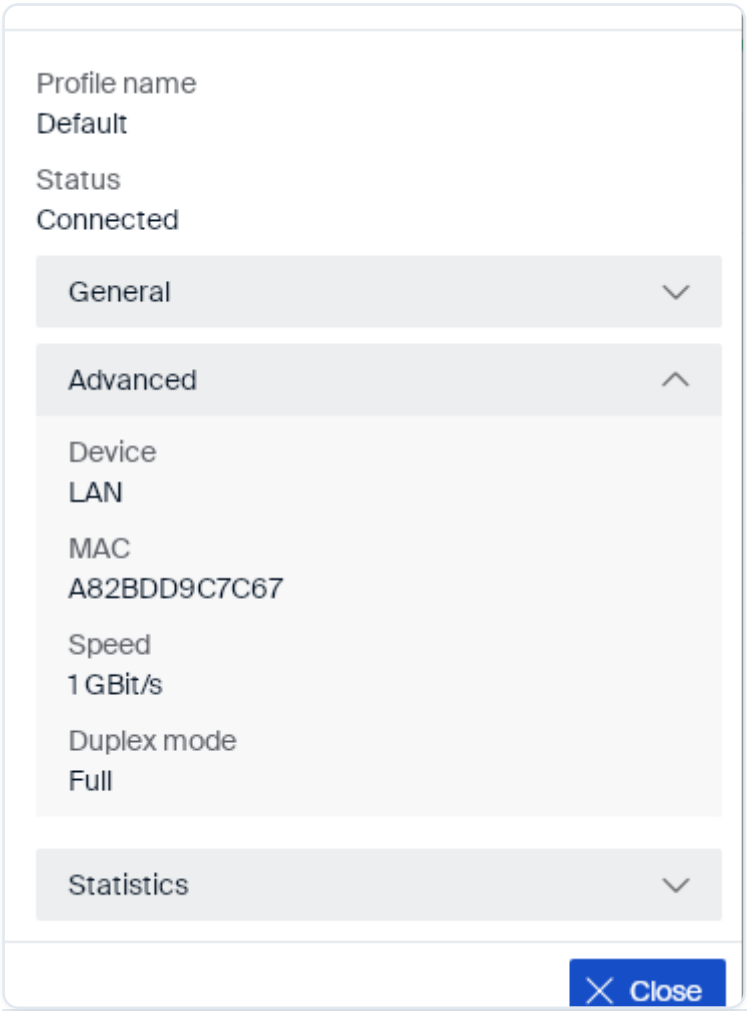
General shows the interface's current addressing information:



FIELD	DESCRIPTION
Address type	How the IP address was obtained, for example `automatic`.
IP address	The interface's current IP address.
Broadcast	The interface's broadcast address.
Subnet Mask	The interface's subnet mask.

ADVANCED

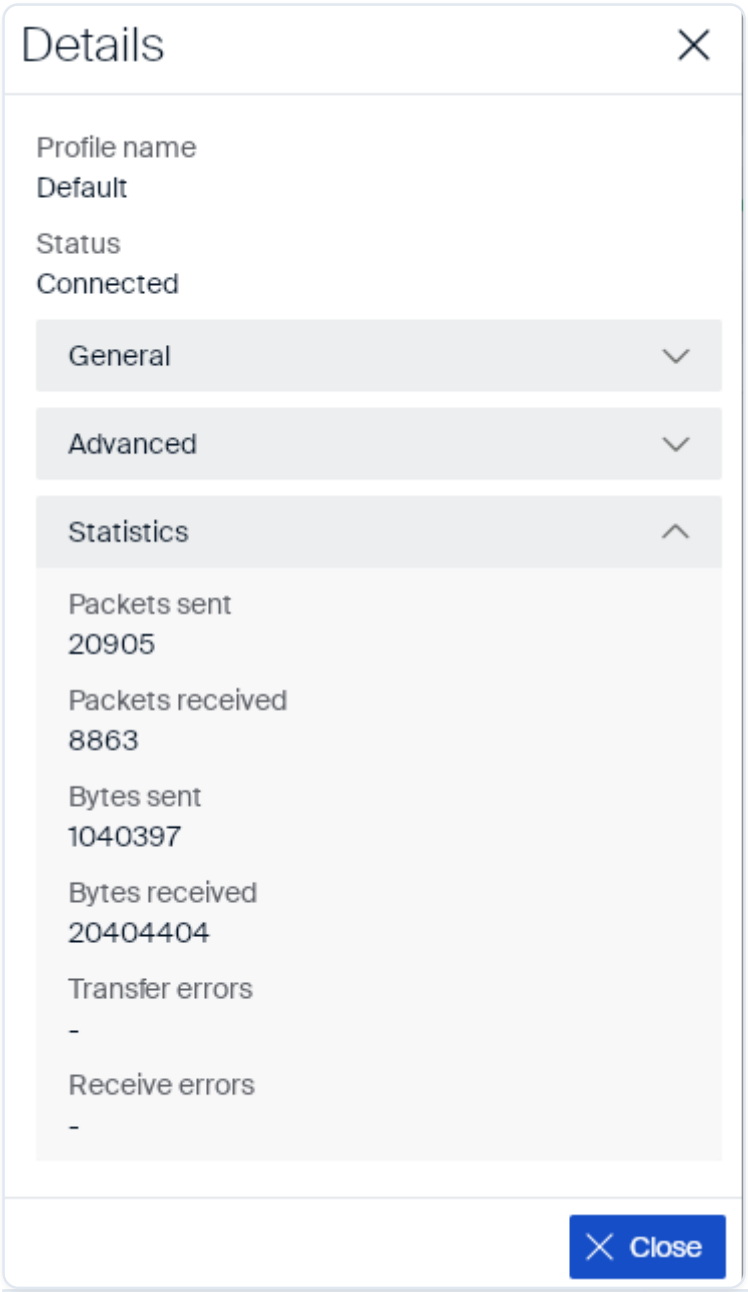
Advanced shows the interface's hardware and link information:



FIELD	DESCRIPTION
Device	The interface type, for example `LAN`.
MAC	The interface's MAC address.
Speed	The interface's negotiated link speed, for example `1 GBit/s`.
Duplex mode	The interface's duplex mode, for example `Full`.

STATISTICS

Statistics shows traffic counters for the interface:



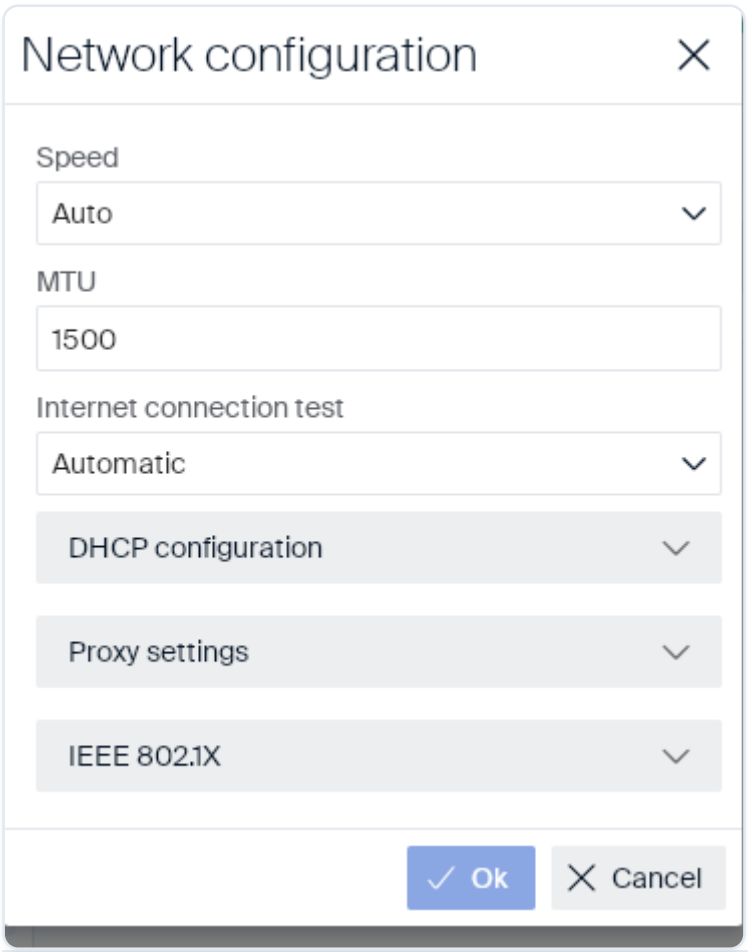
FIELD	DESCRIPTION
Packets sent	Number of packets sent over the interface.
Packets received	Number of packets received over the interface.
Bytes sent	Number of bytes sent over the interface.
Bytes received	Number of bytes received over the interface.
Transfer errors	Number of errors while sending. Shows `-` if none occurred.

FIELD	DESCRIPTION
Receive errors	Number of errors while receiving. Shows `-` if none occurred.

Click **Close** to close the dialog.

Editing an interface's connection settings

Click the wrench icon next to **Active connection**, or click an interface and then **Edit**, to open the **Network configuration** dialog:

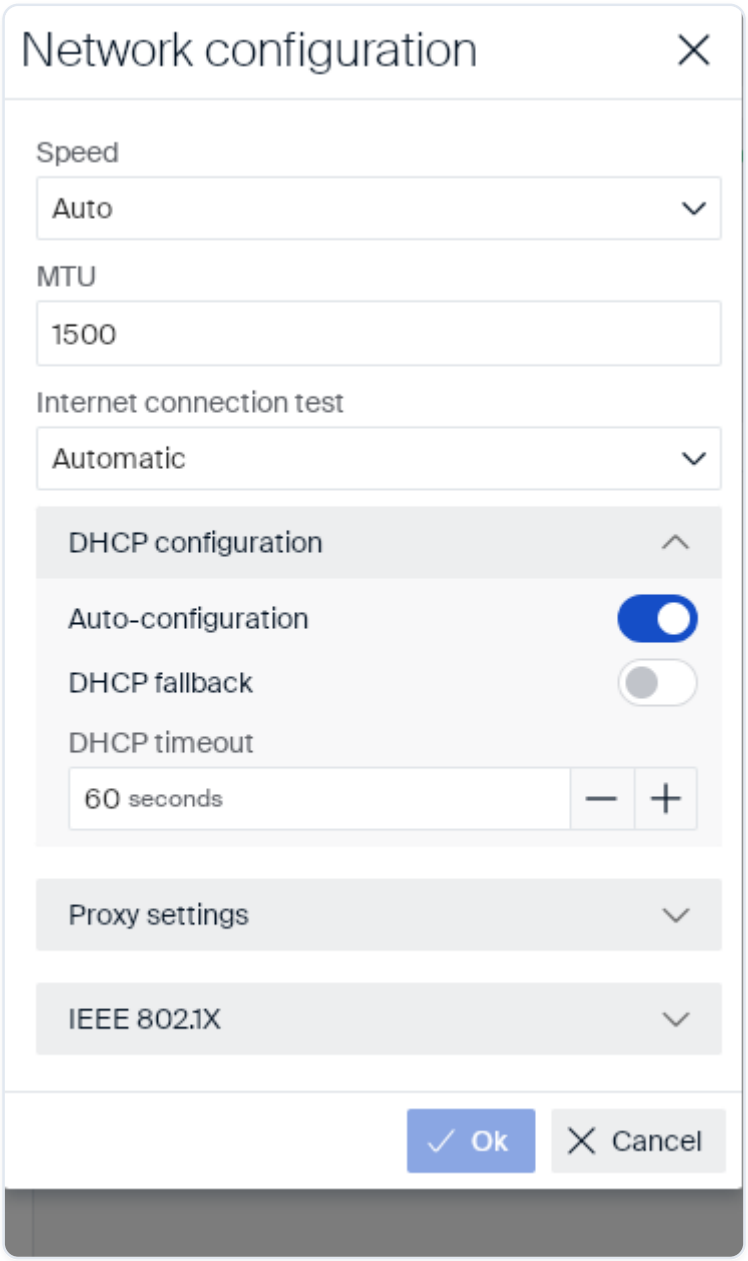


SETTING	TYPE	DESCRIPTION
Speed	Dropdown	The interface's link speed, for example `Auto`.
MTU	Number field	The interface's maximum transmission unit, in bytes.

SETTING	TYPE	DESCRIPTION
Internet connection test	Dropdown	How the device tests for internet connectivity, for example `Automatic`.

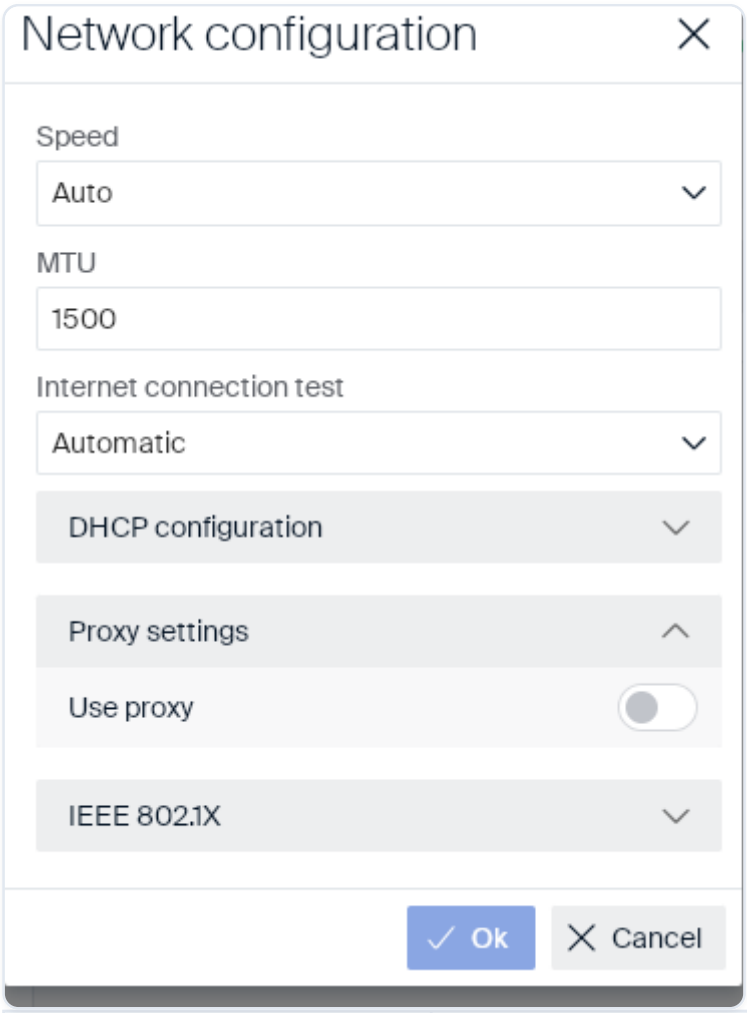
Below these settings, three collapsible sections group further options: **DHCP configuration**, **Proxy settings**, and **IEEE 802.1X**. Click a section's header to expand or collapse it.

DHCP CONFIGURATION



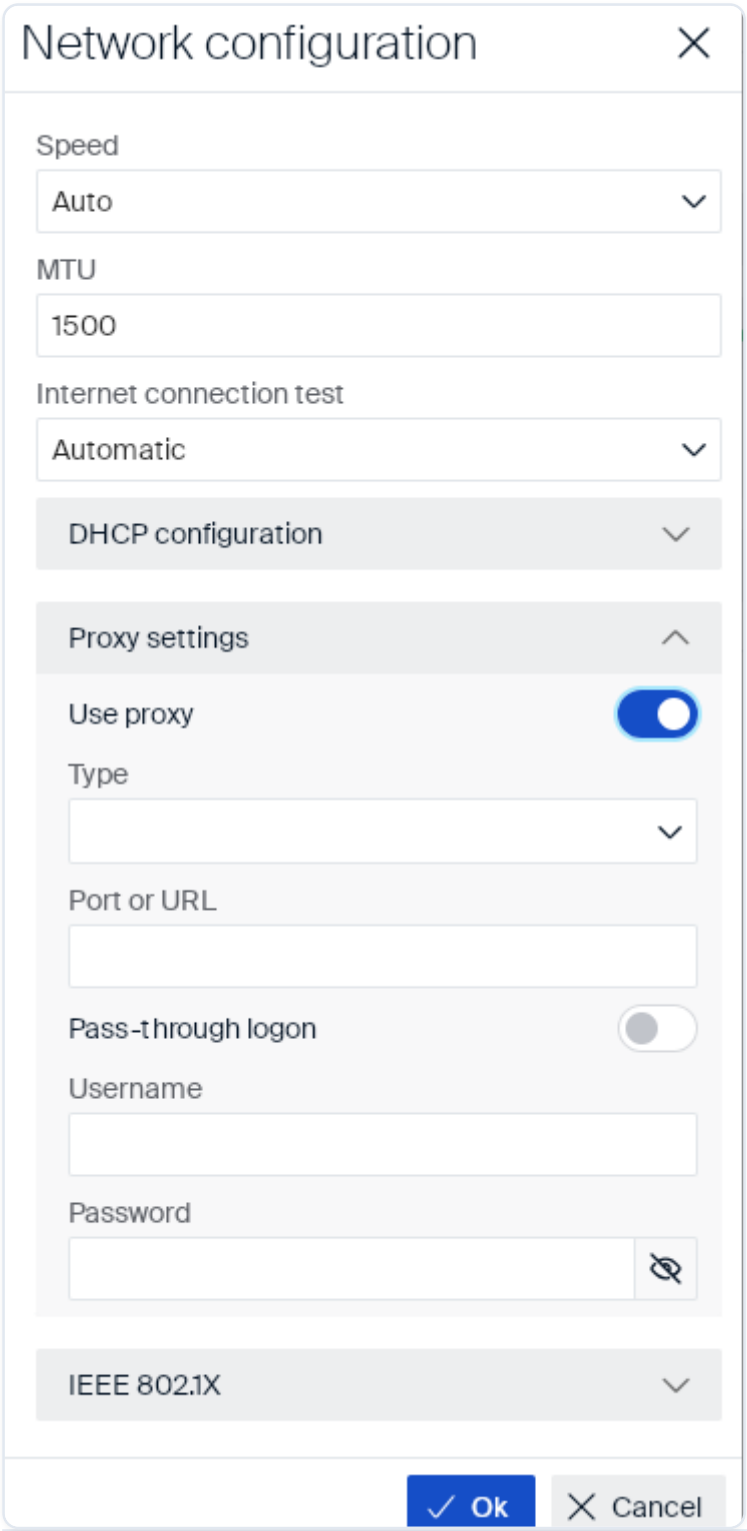
SETTING	TYPE	DESCRIPTION
Auto-configuration	Toggle	Obtains the IP address and network settings automatically over DHCP. Disabling it shows the **IP address**, **Domain**, **Subnet Mask**, **Gateway**, and **DNS servers** fields to set a static configuration instead.
DHCP fallback	Toggle	If DHCP fails on startup, uses the last known settings as long as they're still valid within their lease. Only shown when **Auto-configuration** is enabled.
DHCP timeout	Number field	How long, in seconds, the device waits for a DHCP response. Only shown when **Auto-configuration** is enabled.

PROXY SETTINGS



SETTING	TYPE	DESCRIPTION
Use proxy	Toggle	Enables an HTTP proxy for this interface. Enabling it shows the **Type**, **Port or URL**, and **Pass-through logon** fields described below.

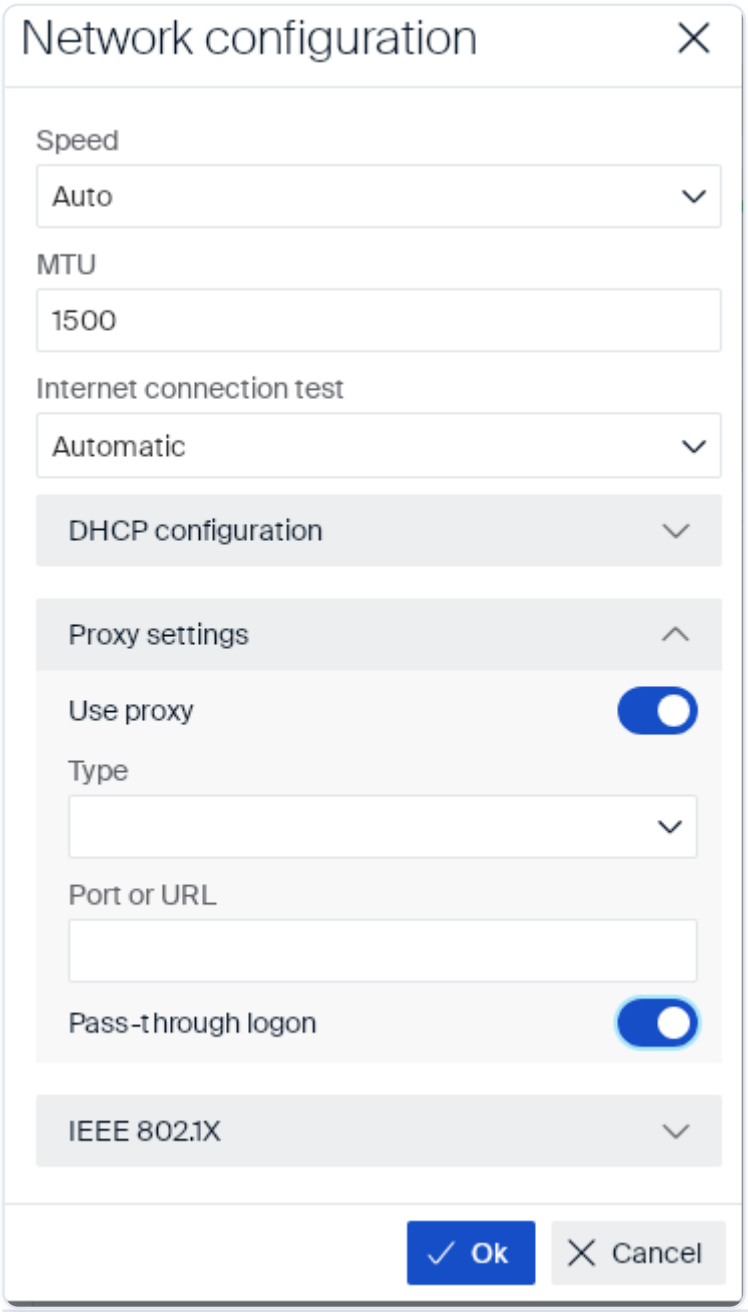
Enabling **Use proxy** shows the following fields:



SETTING	TYPE	DESCRIPTION
Type	Dropdown	The proxy's type.
Port or URL	Text field	The proxy's port or URL.
Pass-through logon	Toggle	Disabling it shows the **Username** and **Password**

SETTING	TYPE	DESCRIPTION
		fields below, used to authenticate with the proxy.

Disabling **Pass-through logon** shows the following fields:



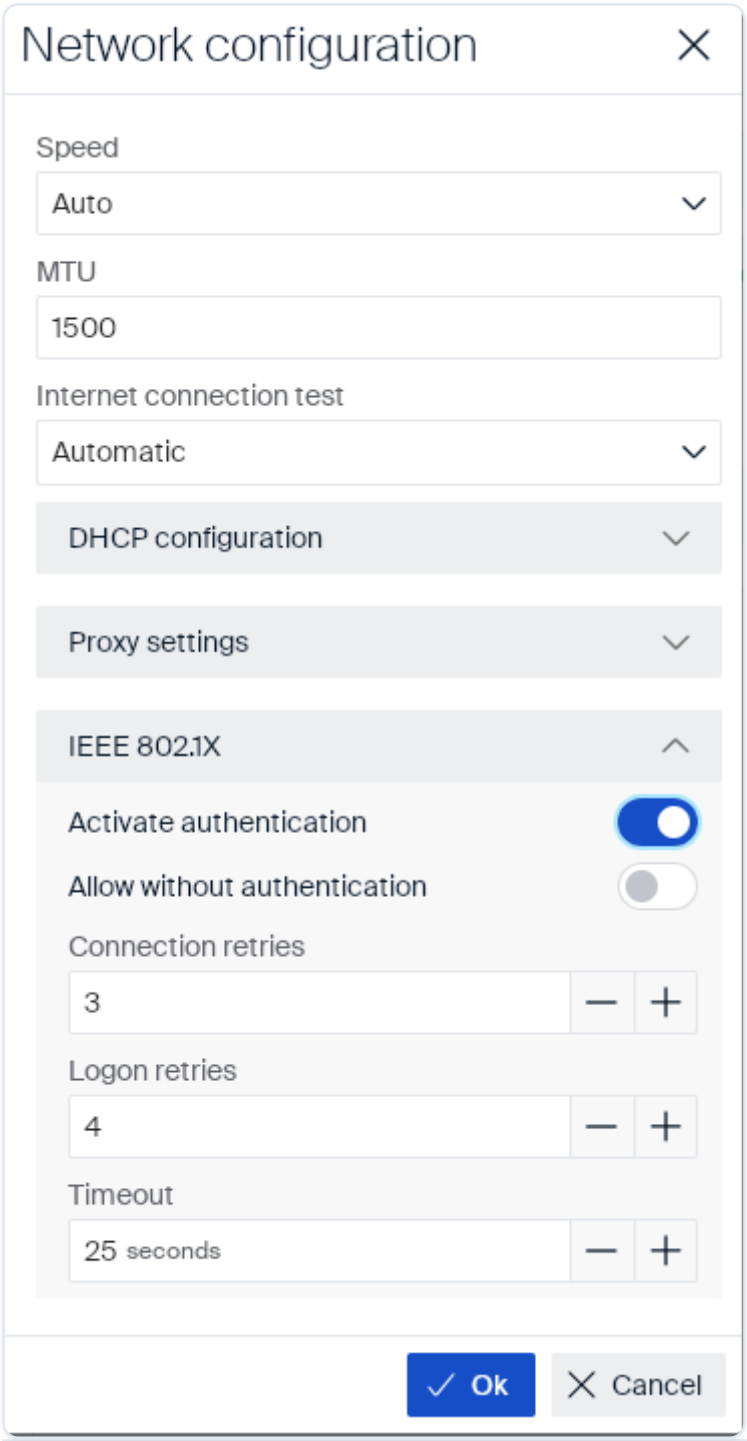
SETTING	TYPE	DESCRIPTION
Username	Text field	The username to authenticate with the proxy.

SETTING	TYPE	DESCRIPTION
Password	Text field	The password to authenticate with the proxy. Click the eye icon to show or hide it.

IEEE 802.1X

SETTING	TYPE	DESCRIPTION
Activate authentication	Toggle	Enables 802.1X network authentication for this interface. Enabling it shows the fields described below.

Enabling **Activate authentication** shows the following fields:



SETTING	TYPE	DESCRIPTION
Allow without authentication	Toggle	Allows the connection without 802.1X authentication.
Connection retries	Number field	How many times the device retries the connection.
Logon retries	Number field	How many times the device retries the logon.

SETTING	TYPE	DESCRIPTION
Timeout	Number field	How long, in seconds, the device waits for authentication to complete.

Click **Ok** to apply the changes, or **Cancel** to discard them.

Using Ethernet and Wi-Fi at the same time

By default, the device uses either Ethernet or Wi-Fi, never both at the same time. If you plug in a LAN cable while the device is connected to Wi-Fi, it switches to the wired connection and disconnects Wi-Fi. If you connect to a Wi-Fi network while the wired connection is up, the wired connection is disconnected.

An administrator can allow parallel connections instead. Ethernet and Wi-Fi then stay connected at the same time, and so can mobile internet. For example, you can connect to a Wi-Fi network while the LAN cable is plugged in. A laptop on a docking station stays connected to Wi-Fi, so it can go on using Wi-Fi when you undock it. A saved Wi-Fi network that is set to connect automatically also connects when the device starts with a LAN cable plugged in.

While several connections are up, the device sends its network traffic over one of them, the active connection. While the preferred connection is up, it is the active connection. By default, the preferred connection is Ethernet. An administrator can prefer Wi-Fi instead. If the active connection drops, for example because you unplug the LAN cable, the device switches to a connection that is still up. After you unplug the LAN cable, the switch takes about 5 seconds. When the preferred connection is available again, the device switches back to it.

With parallel connections, the device shows every connected network as connected:

- In **Network**, for example, **Wireless networks (WiFi)** and **Wired connection (LAN)** both show **Connected**.
- **Active connection** shows the connection that carries the network traffic.
- The **Details** dialog shows the data of the active connection only. The device doesn't show the IP address, MAC address or speed of the other connections.
- In the system bar and in **Quick system access**, also on the login screen and the lock screen, the network icons and the **Wired (LAN)** and **WiFi** tiles show both connections as connected.

The device detects the sign-in page of a Wi-Fi network (captive portal) only when this network is the active connection, for example after you unplug the LAN cable.

CONFIGURING PARALLEL CONNECTIONS

Parallel connections need UniconOS 2611 or later on the device. There is no setting for them on the device. With UniconOS Management 2611 (Scout 15.2611) and later, you configure them in UniconOS Management Board, in the **Network** section of the device configuration: **Allow parallel connections** turns them on, and **Preferred connection type** selects **Ethernet** or **Wi-Fi**.

With Scout 2609 and earlier, set them as advanced file entries instead, one entry per key. In UniconOS Management Board, you find advanced file entries in the **Advanced file entries** section of the advanced device configuration.

The two keys are in the `[Network]` section of `/setup/terminal.ini`. Parallel connections are off until you set `AllowParallelConnections` to `true`:

KEY	VALUES	DEFAULT	DESCRIPTION
`AllowParallelConnections`	`true` or `false`	`false`	`true` keeps Ethernet and Wi-Fi connected at the same time.
`PreferredMedia`	`ethernet` or `wavelan`	`ethernet`	The connection that carries the network traffic while both are connected. `wavelan` is Wi-Fi. This key takes effect only while `AllowParallelConnections` is `true`.

If a key is missing or empty, or has a value that the device doesn't recognize, the device uses the default. For Wi-Fi, enter `wavelan`. Values such as `wlan` or `wifi` aren't recognized, and Ethernet stays preferred.

Devices read these settings when they start. Restart the devices after you change them.

Using MAC address passthrough

A docking station has its own network adapter, with its own MAC address. Without MAC address passthrough, a docked laptop appears on the network with the MAC address of the docking station. With MAC address passthrough, the network adapter of the docking station uses the MAC

address of the laptop's built-in network adapter instead. Use it if your network identifies devices by their MAC address, for example for network access control or DHCP reservations.

eLux® supports MAC address passthrough for the HP USB-C Dock G5 with these laptops:

- Lenovo ThinkPad L15 Gen4 AMD
- Lenovo ThinkPad L16 Gen1 AMD

There is no setting for MAC address passthrough on the device or in Scout. To use it:

1. In the BIOS setup of the laptop, enable **MAC Address Pass Through**.
2. In **Software** in UniconOS Management Board, add the **MAPT Implementations** package to the image of the devices. Make sure that its feature package **MAPT for Lenovo clients** is selected.

   For how to change an image, see Changing an image in the **UniconOS Management Board** guide.
3. Update the firmware of the devices, so that they install the package. See Device configuration / Firmware in the **UniconOS Management Board** guide.

With the BIOS setting and the package in place, the laptop keeps its MAC address on the network when you dock it, also if you dock it while it's running. The device requests its IP address over DHCP with this MAC address. Scout identifies the device by the same MAC address, docked or undocked.

To check which MAC address the device uses, open the **Details** dialog of the active connection, and look at **MAC** under **Advanced**.

> **Important:**

> Before eLux 7.2611, MAC address passthrough for a Lenovo laptop on the ThinkPad Hybrid USB-C with USB-A Dock (type 40AF) worked without additional packages. From eLux 7.2611 on, it also needs the **MAPT Implementations** package with its feature package **MAPT for Lenovo clients**. If your Lenovo laptops use this dock, add the package to their image before you update them to eLux 7.2611.

Mobile internet and eSIM

Devices with a mobile broadband modem (WWAN modem) can connect to the internet over a mobile network. You manage these connections in **Device configuration**: click **Network** in the list of sections, and then expand **Mobile internet (WWAN)**. There you see the SIM cards and eSIM profiles of the device, connect and disconnect, and change the connection settings.

An eSIM is a SIM chip that is built into the device. Instead of inserting a SIM card, you install the subscription of a mobile operator on this chip. Such a subscription is called an eSIM profile. eLux® lets you add and delete eSIM profiles directly on the device, with the activation code that you get from the operator.

To connect through a SIM card or an eSIM profile, the device also needs a mobile internet profile. A mobile internet profile holds the connection settings, for example the access point name (APN) of the operator. The device stores it in its device configuration, while an eSIM profile is stored on the eSIM chip.

REQUIREMENTS

Check these points before you set up mobile internet or eSIM on a device:

- **Modem**: The device has a mobile broadband modem. To use an eSIM, eLux must also be able to reach the eSIM chip through the modem. This isn't possible on every device. For example, the Dell Latitude 5540 (modem Intel XMM7560) can't use its eSIM, but SIM cards work on it.
- **Software**: The image of the device contains the **Mobile Internet** package. For eSIM, the image also contains the **eSIM** feature package of **Mobile Internet**, and the device runs eLux 7.2611 or later. Without the feature package, the device hides the eSIM functions and reports **eSIM profiles are not supported on this device**. You add packages and feature packages to an image in **Software** in UniconOS Management Board, see Changing an image.
- **Second internet connection**: Adding an eSIM profile, switching to another eSIM profile and deleting an eSIM profile all change the eSIM chip. For these actions, the device must be online over LAN or Wi-Fi. The device reports every change on the eSIM chip to the operator, and this report can't depend on the eSIM connection that is changing.
- **Webcam**: To scan the QR code of the operator, the device needs a webcam. Without a webcam, you type the activation code instead.
- **User rights**: Adding an eSIM profile needs the user right for adding network profiles. Connecting, disconnecting and editing need the user right for changing network profiles, and deleting a mobile internet profile needs the right for deleting them. For managed profiles, the matching rights for managed profiles apply. There's no separate user right for eSIM. See User rights.

THE MOBILE INTERNET (WWAN) SECTION

Network shows **Mobile internet (WWAN)** only if the device has a mobile broadband modem. The header of the section shows whether the device is connected over the mobile network. When you expand the section, each SIM card and each eSIM profile of the device has its own tab:

- **SIM card** stands for a SIM card in a SIM slot. Empty SIM slots don't get a tab.
- **eSIM** stands for an eSIM profile on the eSIM chip. An eSIM chip without eSIM profiles doesn't get a tab, but you can add an eSIM profile to it.

If the device has several SIM cards or several eSIM profiles, their tabs are numbered, for example **SIM card 1** or **eSIM 2**. A green dot on a tab marks the SIM that the device is connected through. If there's no tab at all, the section shows **No SIM card inserted**.

After the device starts, it can take a few minutes until all SIMs are shown. The device first reads its SIM slots one after the other.

The section shows the connection through the SIM of the selected tab, the mobile internet profiles of this SIM, and the buttons for adding eSIM profiles. From top to bottom, you find these elements:

ELEMENT	DESCRIPTION
Operator name	The mobile operator of the SIM. If the operator name isn't available, the modem model is shown instead.
Information	Opens the **Modem & eUICC information** dialog with details about the modem, the eSIM chip and the eSIM profile. See Modem and eSIM chip details.
ACTIVE PROFILE	The mobile internet profile that the device is connected through, or **No active connection**. Next to it, the status of the connection: **Connected**, **Disconnected**, **Connecting**, **Disconnecting**, **Failed to connect** or **Failed to disconnect**. While the device is connected, this area also shows the **IP address** and the **APN** in use, and the buttons **Disconnect** and **Edit**.
ALL PROFILES	The mobile internet profiles of this SIM, with their number in brackets. A profile is marked

ELEMENT	DESCRIPTION
	User profile if it was created on the device, or **Managed profile** if it comes from Scout. The menu icon next to a profile offers **Connect**, **Disconnect**, **Edit** and **Delete**. **Add** creates a new mobile internet profile.
Add eSIM profile via QR code	Adds an eSIM profile from the QR code of the operator, which you scan with the webcam.
Add eSIM profile manually	Adds an eSIM profile from an activation code that you type.

The two buttons for adding an eSIM profile appear only if eLux can reach the eSIM chip of the device, and only for users who are allowed to add network profiles.

Modem and eSIM chip details

Click **Information** on a tab to open the **Modem & eUICC information** dialog. eUICC is the technical term for the eSIM chip. The dialog shows details that the operator or support staff may ask for, and it has the **Delete** button for the eSIM profile.

The dialog groups its details in three sections, one each for the modem, the eSIM chip and the eSIM profile:

SECTION	FIELD	DESCRIPTION
MODEM	**Model**	Which mobile broadband modem the device has.
MODEM	**Device ID**	The system path of the modem on the device.
MODEM	**Primary SIM slot**	The SIM slot that the modem currently uses.
EUICC	**EID**	The identification number of the eSIM chip. Some operators ask for it when they issue an eSIM profile.
EUICC	**SIM slot**	The SIM slot that the eSIM chip is connected to. The modem switches to this slot

SECTION	FIELD	DESCRIPTION
		when the device connects through an eSIM profile.
EUICC	**Firmware version**	The version of the software that runs on the eSIM chip.
EUICC	**APDU backend**	The interface that the device uses to communicate with the eSIM chip. Only relevant for support cases.
PROFILE	**Name**	The name of the eSIM profile, as defined by the operator.
PROFILE	**Service provider**	The mobile operator of the eSIM profile.
PROFILE	**ICCID**	The identification number of the eSIM profile. The device uses it to assign mobile internet profiles to this eSIM profile.
PROFILE	**Enabled**	**Yes** if the eSIM chip currently uses this eSIM profile.

ADDING AN ESIM PROFILE

To add an eSIM profile, you need its activation code from the mobile operator. Operators usually supply it as a QR code. As text, an activation code looks like `LPA:1$operator.com$ABCDE-12345`.

1. In **Device configuration**, click **Network** in the list of sections, and then expand **Mobile internet (WWAN)**.
2. Enter the activation code in one of these ways:
   - To scan the QR code, click **Add eSIM profile via QR code**. If the device has several webcams, select one under **Device**. Then hold the QR code in front of the webcam, well lit and fully visible. As soon as the device recognizes an activation code, the **Add eSIM profile** dialog opens with the code filled in.

- To type the code, click **Add eSIM profile manually**. In the **Add eSIM profile** dialog, type the code in **Activation Code**.

3. Click **Continue** to start the installation. After a QR code scan, the installation starts by itself if the device is online.

   If the device isn't online over LAN or Wi-Fi, **Continue** is unavailable, and the dialog shows **An active internet connection is required to install a new eSIM profile.**

4. If the dialog shows a **Confirmation code** field, type the confirmation code from the operator, and click **Continue**. Some operators protect their eSIM profiles with this additional code.
5. Wait while the dialog shows **Installing eSIM profile…**.

   The device downloads the eSIM profile from the operator and writes it to the eSIM chip. This can take up to five minutes. Don't turn off the device in the meantime.

6. Select the mobile internet profile that the device should use with the new eSIM profile. To set up a new one, select **Create a new WWAN configuration**, enter the **APN** of the operator, and click **Ok**.
7. Click **Connect** to connect through the new eSIM profile, or **Close** to connect later.

The new eSIM profile gets its own **eSIM** tab. If you clicked **Connect**, the device connects through it, and the tab shows a green dot.

If you selected a mobile internet profile that already belongs to another SIM, the device creates a copy of it for the new eSIM profile. The original profile stays with its SIM.

If adding the eSIM profile fails, the dialog stays open, so you can try again. Your other eSIM profiles aren't affected. The message in the dialog tells you the cause and what to do:

MESSAGE	WHAT TO DO
The scanned QR code is not an eSIM activation code.	The QR code contains something else, for example a web address. Scan the QR code with the activation code from the operator.
This activation code is invalid or has already been used.	Check the code. If it was already used, for example for an eSIM profile that was deleted later, ask the operator for a new activation code.
This eSIM profile is already installed on this device.	No action needed. The eSIM profile is already on the eSIM chip of this device.

MESSAGE	WHAT TO DO
Confirmation code is required	Type the confirmation code from the operator in **Confirmation code**.
The confirmation code was refused. Please check it and try again.	Type the confirmation code from the operator again.
The operator's server could not be reached.	Check the internet connection of the device over LAN or Wi-Fi. If the device uses a proxy, see Using a proxy.

CONNECTING THROUGH A SIM CARD OR AN ESIM PROFILE

The modem can use only one SIM at a time, even if the device has a SIM card and an eSIM. The eSIM chip can hold several eSIM profiles, but it also uses only one of them at a time. You don't need to switch SIM slots or activate eSIM profiles yourself. When you connect through a mobile internet profile, the device switches the modem to the SIM of this profile. For an eSIM profile, the device also activates it on the eSIM chip, which deactivates the eSIM profile used before.

1. In **Mobile internet (WWAN)**, click the tab of the SIM card or eSIM profile that you want to use.
2. Under **ALL PROFILES**, click the menu icon next to the mobile internet profile, and then click **Connect**.

The device switches to the SIM and connects, which can take a few seconds. A connection through another SIM ends. When the connection is established, the tab shows a green dot, and **ACTIVE PROFILE** shows **Connected**, the IP address and the APN.

If **Connect** fails for an eSIM profile, make sure that the device is online over LAN or Wi-Fi, see Requirements.

To end the connection, click **Disconnect** under **ACTIVE PROFILE**. The eSIM profile stays active on the eSIM chip, so you can connect again right away.

ADDING AND EDITING MOBILE INTERNET PROFILES

To create a mobile internet profile, open the tab of the SIM, and click **Add** under **ALL PROFILES**. To change a profile, click the menu icon next to it, and then click **Edit**. For the profile in use, you can also click **Edit** under **ACTIVE PROFILE**. In each case, the **Network configuration** dialog opens.

The dialog holds the name of the profile, how the device connects, and the access data from the operator:

SETTING	TYPE	DESCRIPTION
Profile name	Text field	The name of the profile in the **ALL PROFILES** list. To rename the profile, change the name here.
Connect automatically	Toggle	The device connects through this profile by itself, if the conditions in Connecting automatically are met.
Roaming	Toggle	Keeps the connection when the device is outside the network of its operator.
PIN	Text field	The PIN of the SIM. If you leave it empty and the SIM needs a PIN, the device asks for it when it connects. Click the eye icon to show or hide it.
APN	Text field	The access point name from the operator. Required.
Username	Text field	The user name of the mobile account, if the operator requires one.
Password	Text field	The password of the mobile account, if the operator requires one. If you leave it empty and the operator requires it, the device asks for it when it connects. Click the eye icon to show or hide it.
Internet connection test	Dropdown	How the device tests whether the connection reaches the internet, for example `Automatic`.

Click **Ok** to save the profile, or **Cancel** to discard your changes.

> **Note:**

> A PIN or password that you type when the device asks for it is saved in the profile as soon as the connection succeeds. The device then doesn't ask again. A wrong PIN is never saved. Otherwise, the device would try it again and again until the SIM asks for the PUK. For a managed profile, the device doesn't save what you type.

> To delete a mobile internet profile, click the menu icon next to it, and click **Delete**. Confirm the message **Do you want to delete the network profile "…"?** This deletes only the settings. An eSIM profile that the mobile internet profile used stays on the eSIM chip, and you can use it with another mobile internet profile.

DELETING AN ESIM PROFILE

Delete an eSIM profile only if the subscription itself should leave the device. To stop using a mobile connection, disconnect it, or delete the mobile internet profile instead.

> **Important:**

> You can't undo the deletion. To install the eSIM profile again, you normally need a new activation code from the operator, because an activation code can usually be used only once. Some operators accept a code again, so check with your operator.

1. Open the **eSIM** tab of the eSIM profile, and click **Information**.
2. In the **PROFILE** section of the **Modem & eUICC information** dialog, click **Delete**.

   If the device isn't online over LAN or Wi-Fi, **Delete** is unavailable, and the dialog shows **An active internet connection, not provided by this eSIM profile, is required to remove it.**
3. Confirm the message **Do you want to remove the eSIM profile "…"?** with **Delete**.

The dialog shows **Deleting eSIM profile…** while the device deletes the eSIM profile from the chip and informs the operator. If the eSIM profile is active, the device deactivates it first. Then the **eSIM** tab of the eSIM profile disappears. Mobile internet profiles that used the eSIM profile stay, but they can't connect anymore.

A factory reset, a recovery installation or a firmware update doesn't delete eSIM profiles, because they're stored on the eSIM chip. The mobile internet profiles, however, are part of the device configuration and are reset with it. To remove an eSIM profile from a device for good, for example before the device is taken out of service, delete it as described above.

CONNECTING AUTOMATICALLY

If **Connect automatically** is on for a mobile internet profile, the device connects through it by itself, but only if all of these conditions are met:

- No LAN cable is connected to the device.
- The profile is assigned to a SIM, see Mobile internet profiles from Scout.
- This SIM is the one that the modem currently uses. An automatic connection never switches the SIM. Only **Connect** does.

After a restart, the device therefore connects automatically through the SIM that it used last. Turn on **Connect automatically** only for the profiles of one SIM. If profiles of different SIMs have it turned on, you can't predict which one connects.

MOBILE INTERNET PROFILES FROM SCOUT

You can also define mobile internet profiles centrally in UniconOS Management Board, in the **Mobile Internet** group of the **Network** section of the device configuration. On the device, these profiles are marked **Managed profile**. eSIM profiles, however, can only be added and deleted on the device, because adding one needs the activation code.

A mobile internet profile from Scout doesn't say which SIM it's for. The device assigns it to a SIM when it reads its SIMs:

- If the device has exactly one SIM card or eSIM profile, the device assigns the profile to it.
- If the device has more than one, for example a SIM card and an eSIM profile, the device doesn't choose. The profile stays unassigned and doesn't connect. If **Connect automatically** is on for the profile, the device shows this notification with the title **Mobile broadband**: **This device holds more than one SIM. Choose which one the mobile broadband profile '…' connects through.**

As long as the device hasn't read its eSIM chip yet, for example shortly after it starts, it counts the eSIM chip as more than one SIM, because the chip can hold several eSIM profiles. After the device has read the chip, it can assign the profile by itself.

To assign the profile, open the tab of the SIM that it should use, and connect through the profile there, see Connecting through a SIM card or an eSIM profile. If the connection fails, the profile stays unassigned. The device keeps the assignment, also after a restart.

If you roll out a mobile internet profile from Scout to devices with a SIM card and an eSIM, plan for this step on each device.

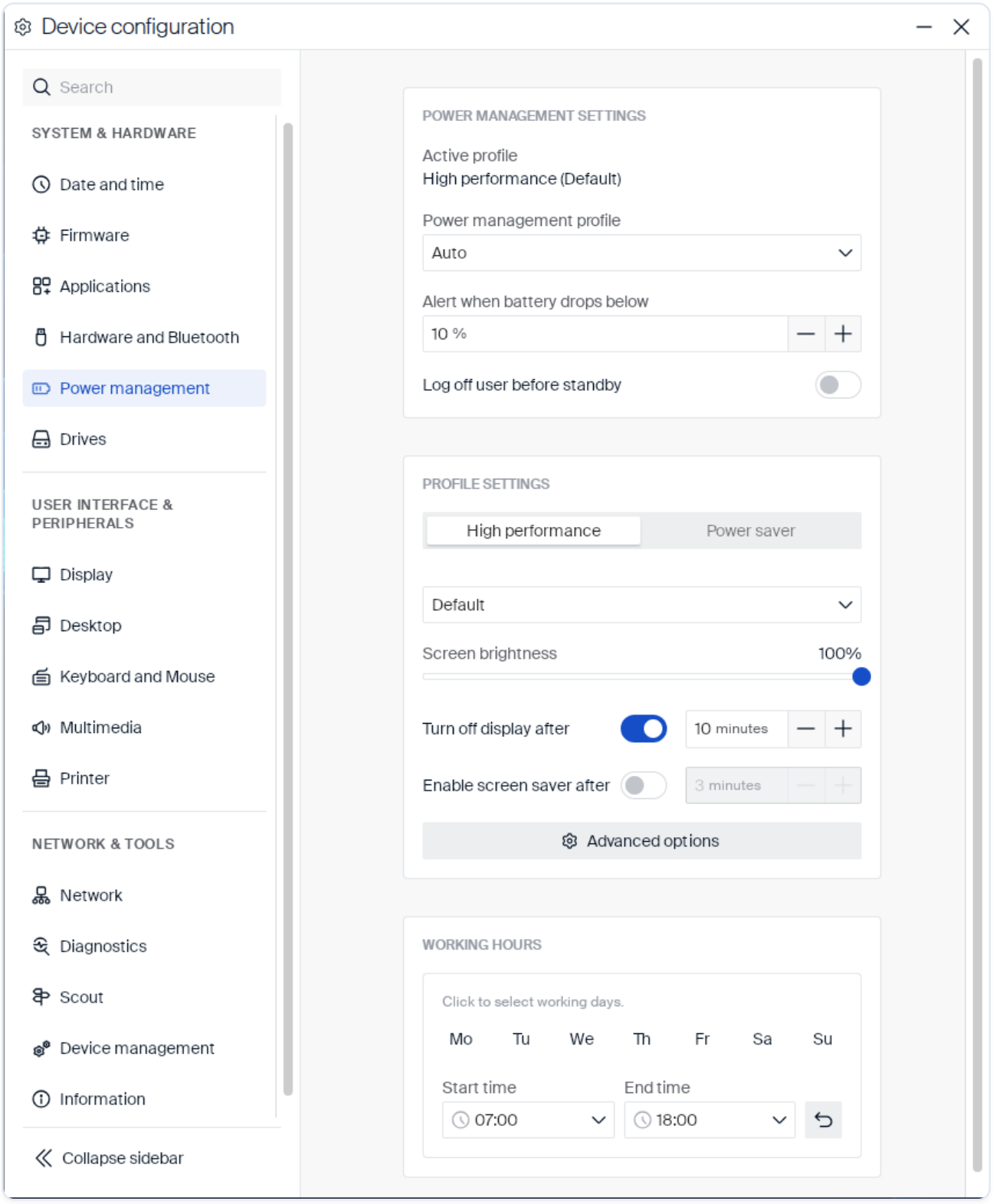
USING A PROXY

If the device reaches the internet through a proxy, it also uses the proxy when it adds or deletes an eSIM profile, or switches to another eSIM profile. For this, the image of the device must contain the **Authenticated Network Proxy** feature package. Without it, adding an eSIM profile fails with **The operator's server could not be reached.**

For the proxy settings of a network connection, see Proxy settings.

Power management

Power management is where you configure energy-saving settings, including battery alerts and standby behavior on devices that have a battery.

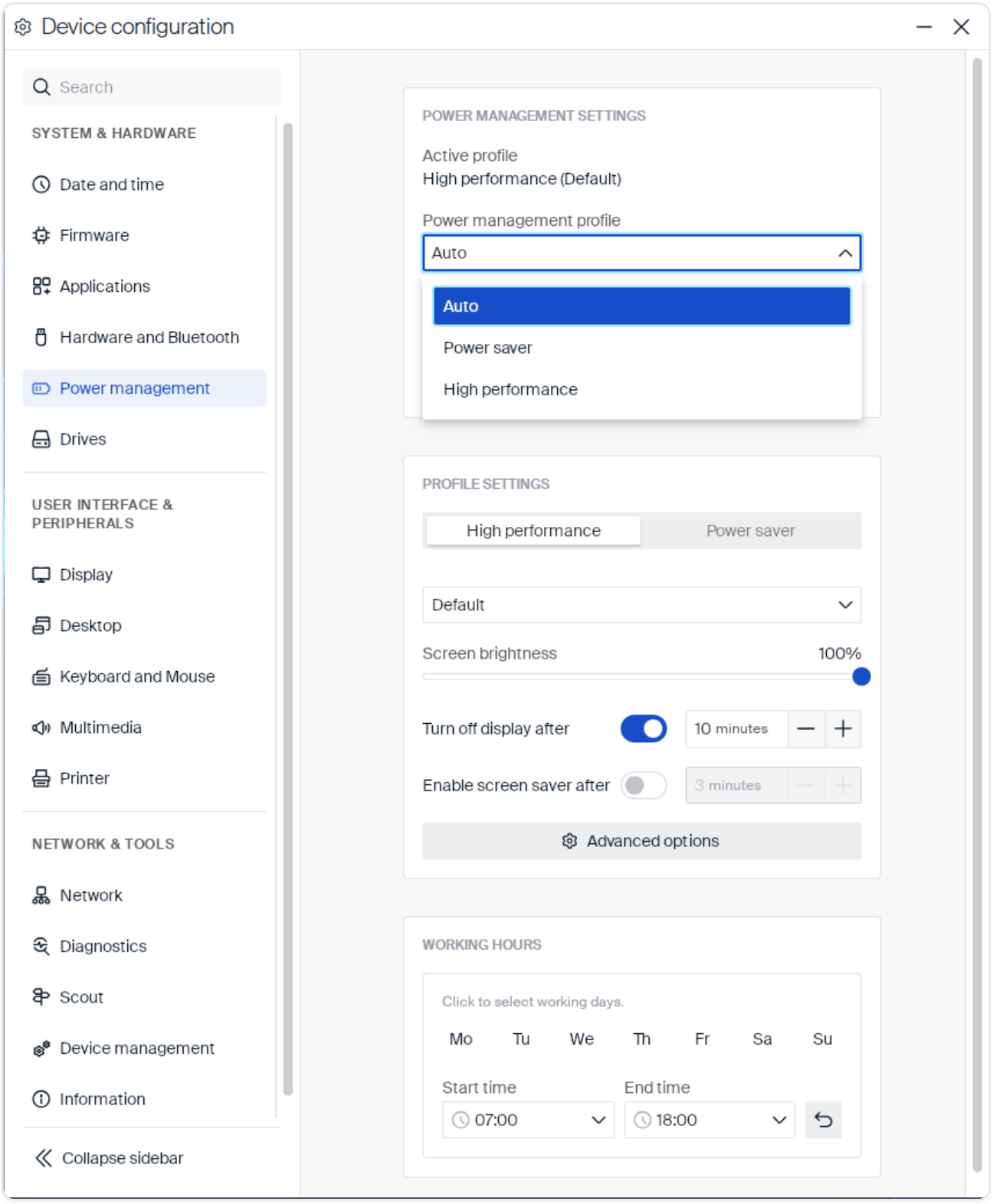


Power management settings

These settings apply regardless of which profile is active — they control the overall power mode and what happens on low battery:

SETTING	TYPE	DESCRIPTION
Active profile	Info	The profile currently in effect, followed by its active preset in parentheses, for example `High performance (Default)`. Read-only.
Power management profile	Dropdown	`Auto`, `Power saver`, or `High performance`. `Auto` switches to **Power saver** while the device runs on battery power, and to **High performance** while it's connected to AC power.
Alert when battery drops below	Number field	Battery percentage at which the device shows a low-battery alert. Adjust it with the **-/+** buttons, in steps of 5%.
Log off user before standby	Toggle	Logs off the current user before the device enters standby.

Click **Power management profile** to see its options:

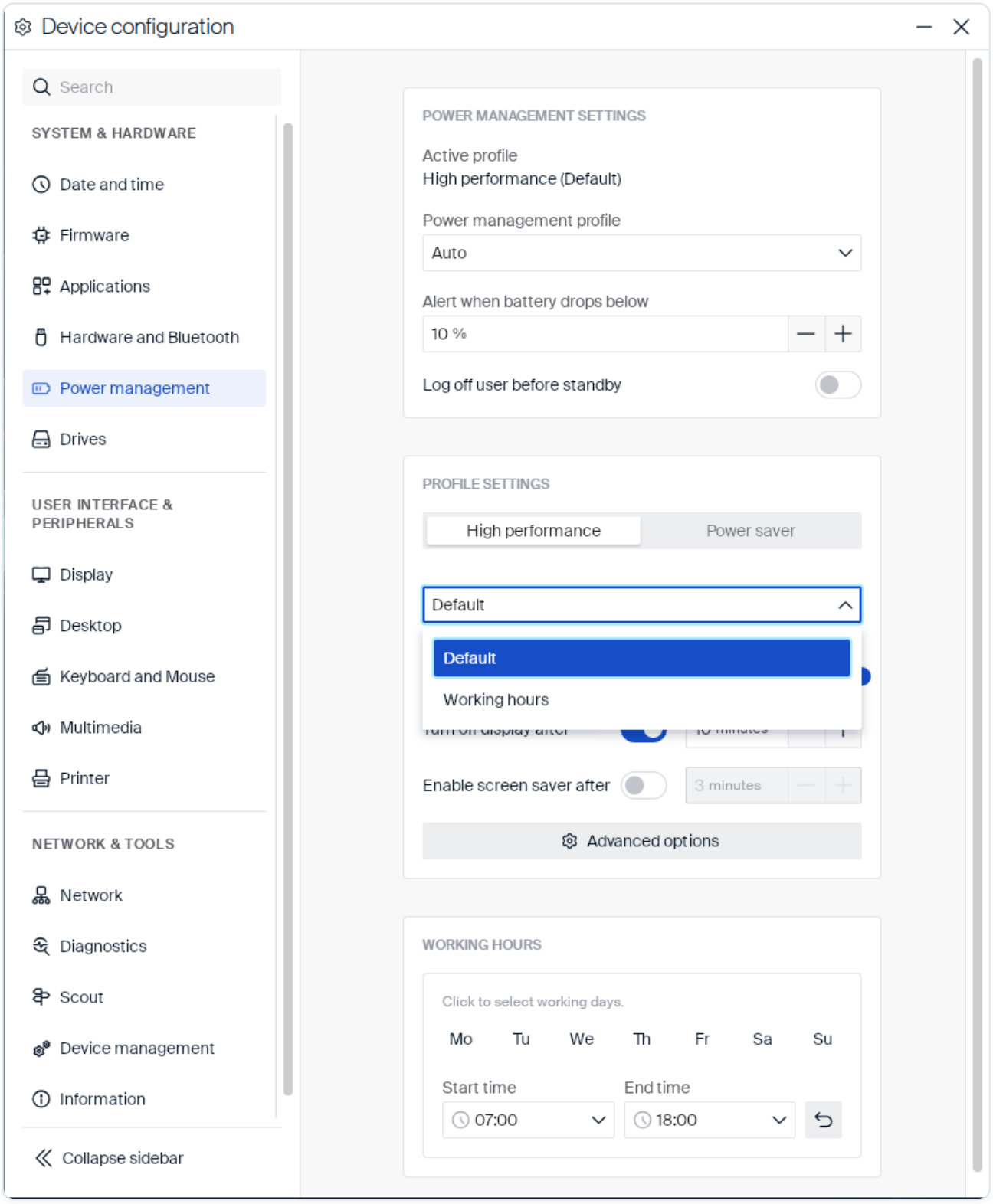


Profile settings

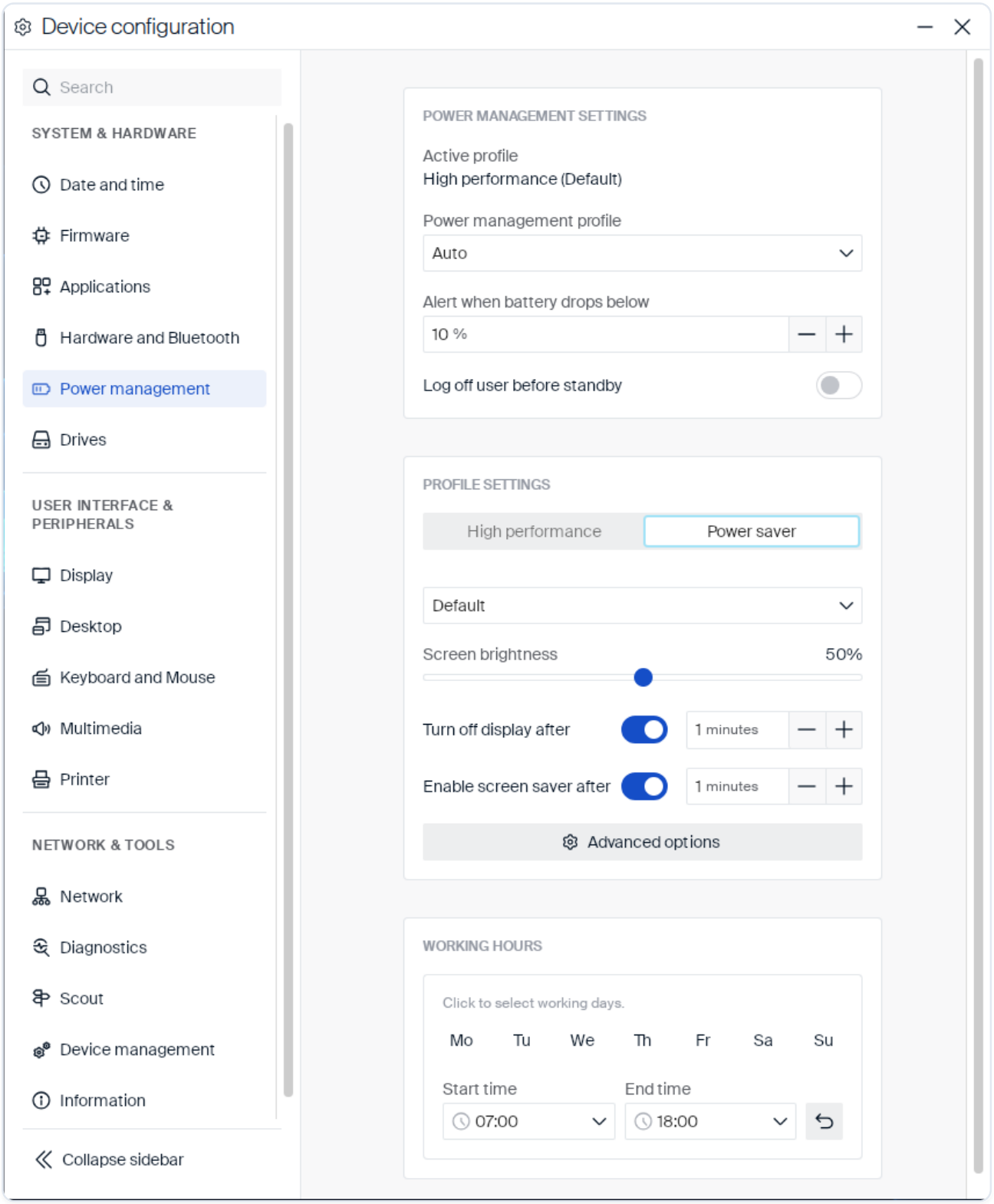
Each profile (**High performance** and **Power saver**) has its own display and screen saver settings. Select the tab for the profile you want to configure:

SETTING	TYPE	DESCRIPTION
Preset	Dropdown	`Default` or `Working hours`. The device uses the **Working hours** preset during the days and times set under **Working hours** below, and the **Default** preset at all other times. If no working days are selected, **Default** always applies.
Screen brightness	Slider	Sets the display brightness while this profile and preset are active.
Turn off display after	Toggle + number field	Turns off the display after the device is idle for the given number of minutes.
Enable screen saver after	Toggle + number field	Starts the screen saver after the device is idle for the given number of minutes.

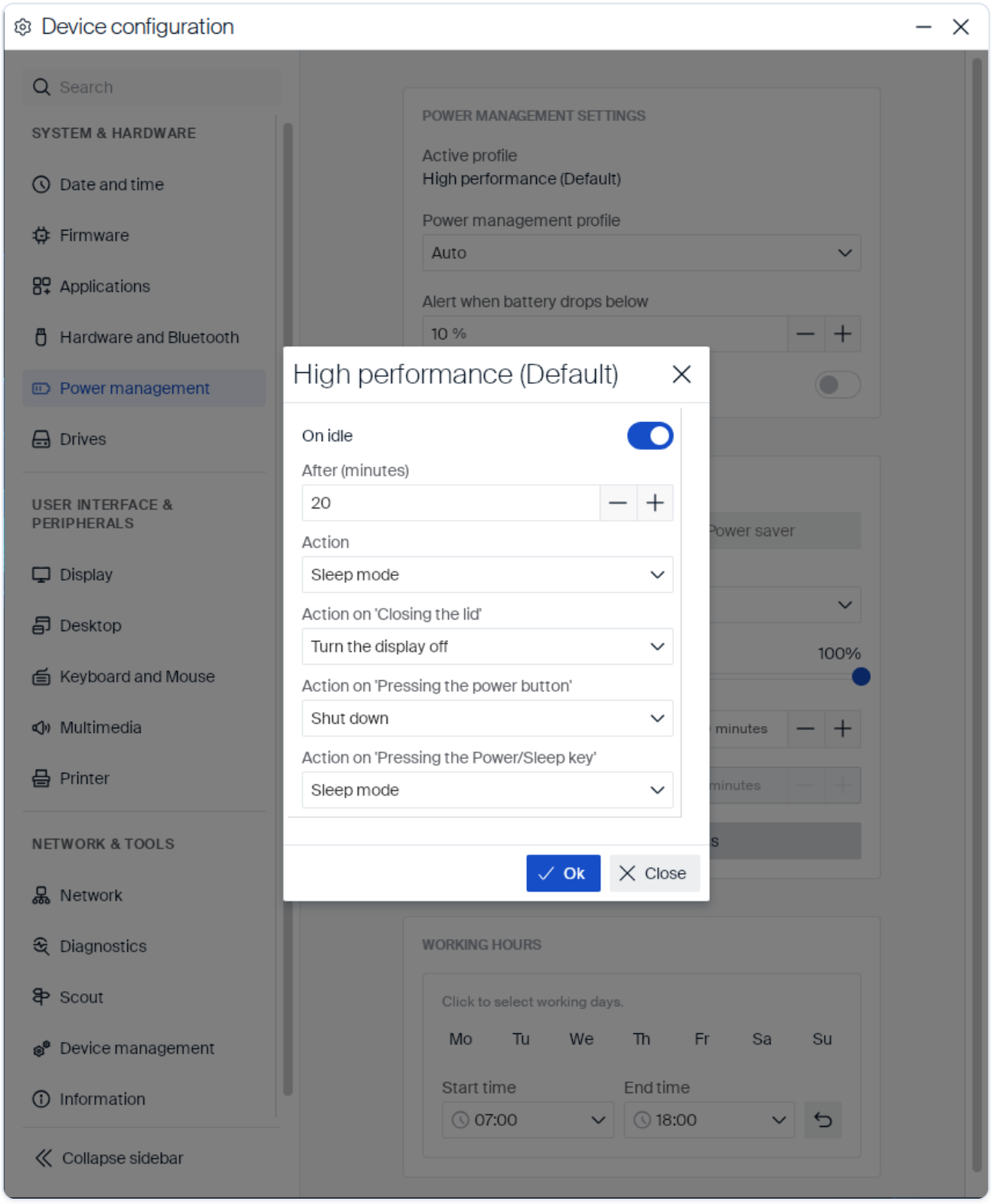
Click **Preset** to see its options:



For example, the **Power saver** tab with its **Default** preset selected:



Click **Advanced options** for further settings, specific to the selected profile and preset:



The dialog title names the profile and preset it configures, for example **High performance (Default)**:

SETTING	TYPE	DESCRIPTION
On idle	Toggle	Enables an automatic action after the device is idle for a set time. Disabling it grays

SETTING	TYPE	DESCRIPTION
		out **After (minutes)** and **Action** below, but the actions for the lid, power button, and Power/Sleep key still apply regardless.
After (minutes)	Number field	How many minutes of inactivity trigger **Action**.
Action	Dropdown	The action to take on idle: `Shut down`, `Sleep mode`, `Restart`, or `Log off`.
Action on 'Closing the lid'	Dropdown	The action to take when the lid is closed: `No action`, `Turn the display off`, `Shut down`, or `Sleep mode`.
Action on 'Pressing the power button'	Dropdown	The action to take when the power button is pressed: `No action`, `Turn the display off`, `Shut down`, or `Sleep mode`.
Action on 'Pressing the Power/Sleep key'	Dropdown	The action to take when the Power/Sleep key is pressed: `No action`, `Shut down`, or `Sleep mode`.

Click **Ok** to apply the changes, or **Close** to discard them.

Working hours

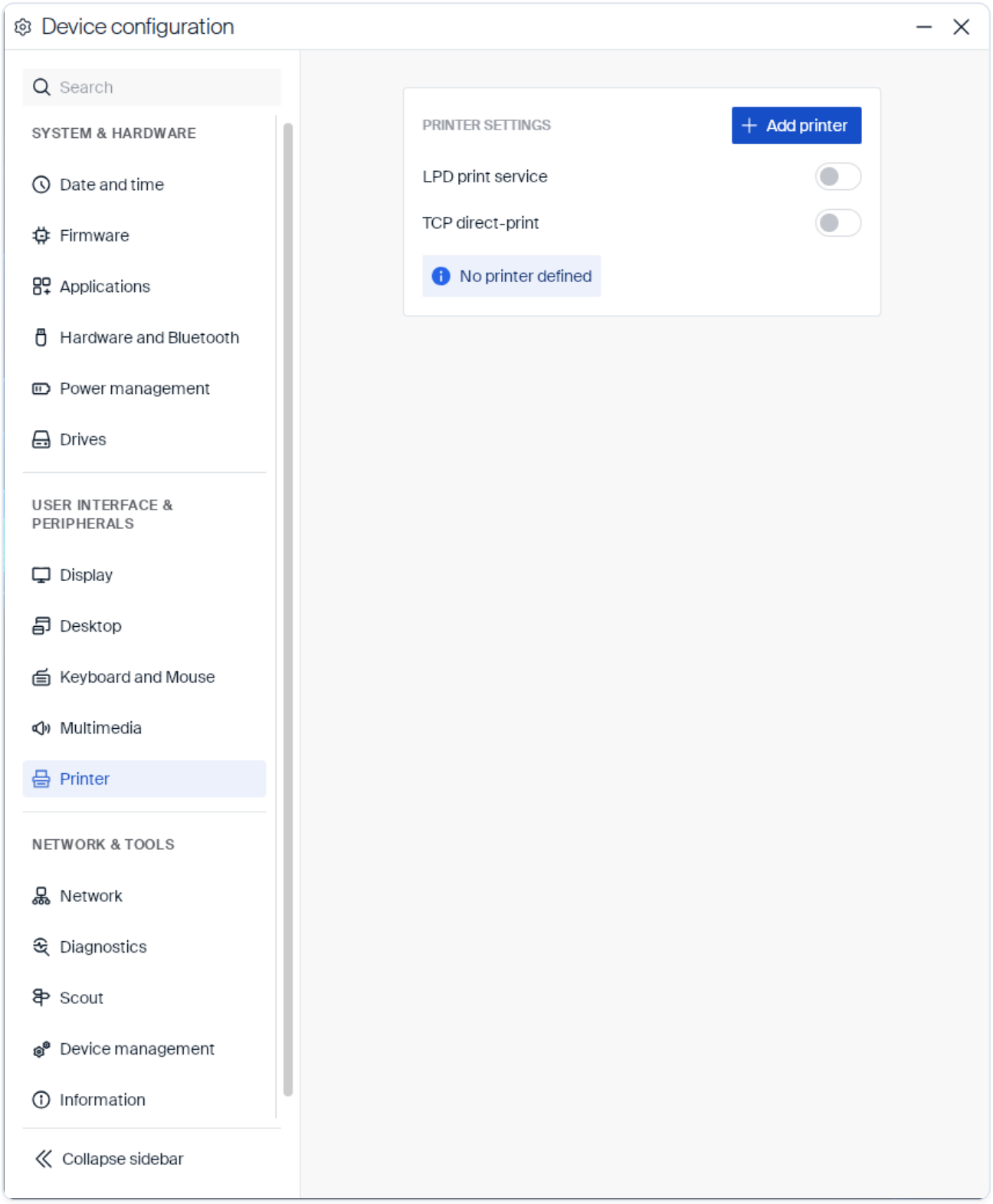
Working hours applies to both profiles. During the days and times you set here, each profile uses its **Working hours** preset instead of its **Default** preset:

1. Click the days of the week the device is normally in use.
2. Set **Start time** and **End time** for those working hours. Type a time directly, or select one from the dropdown list. Times are listed in 30-minute increments, in 24-hour `HH:MM` format.

Click the reset button next to the time fields to clear the selected working days and reset **Start time** and **End time** to `07:00` and `18:00`.

Printer

Printer is where you configure locally connected and network-connected printers.



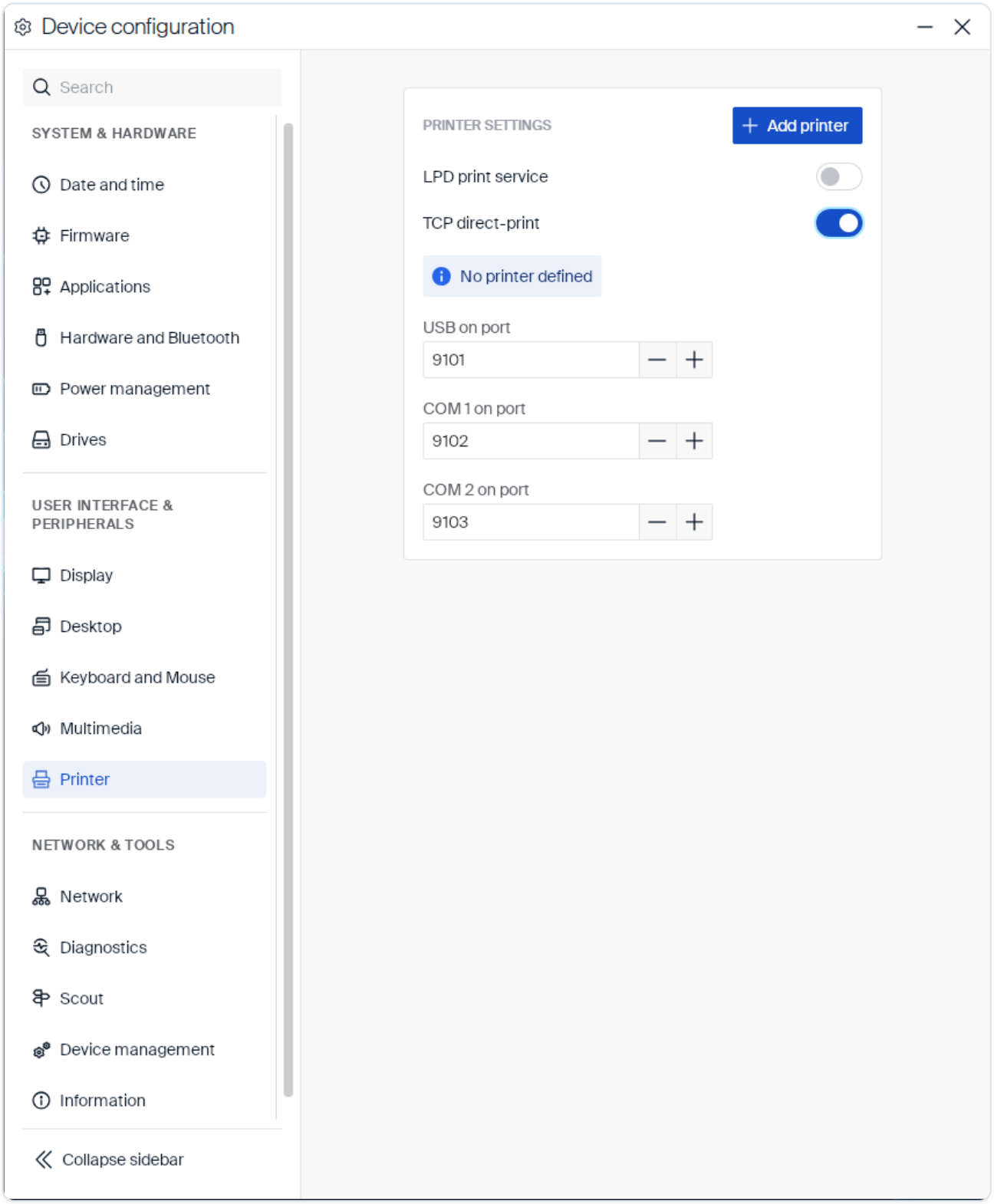
Print services

These toggles turn on services that let the device accept print jobs sent to it over the network:

SETTING	TYPE	DESCRIPTION
LPD print service	Toggle	Enables the Line Printer Daemon service, so the device can accept print jobs sent to it over the network.
TCP direct-print	Toggle	Enables direct TCP printing to a network printer connected to the device.

Turning on **TCP direct-print** shows three number fields, one for each connection it listens on. Adjust each port with its **-/+** buttons:

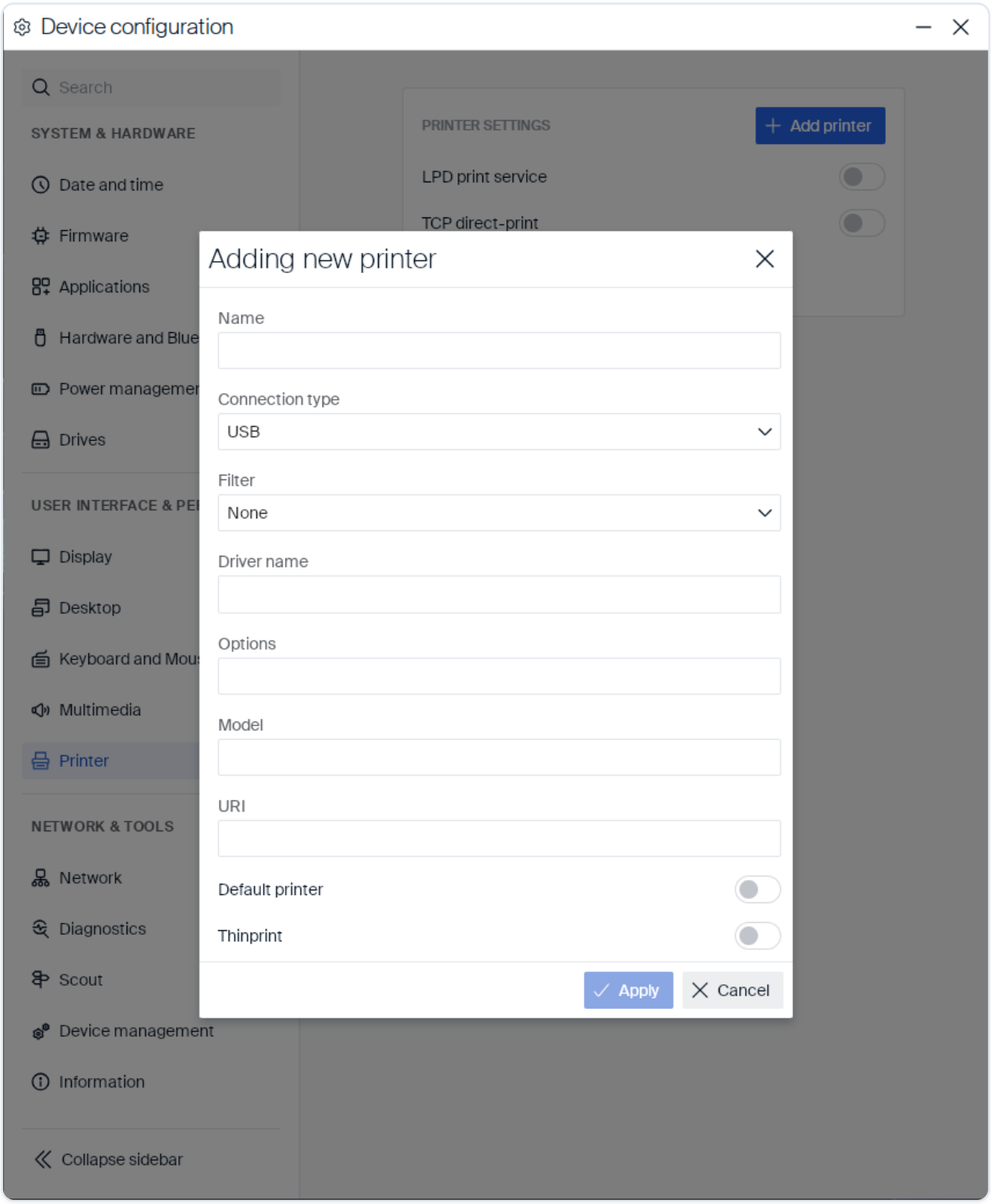
SETTING	TYPE	DEFAULT
USB on port	Number field	`9101`
COM 1 on port	Number field	`9102`
COM 2 on port	Number field	`9103`



While no printer is configured yet, a **No printer defined** message appears below these toggles.

Adding a printer

1. Click **Add printer**.

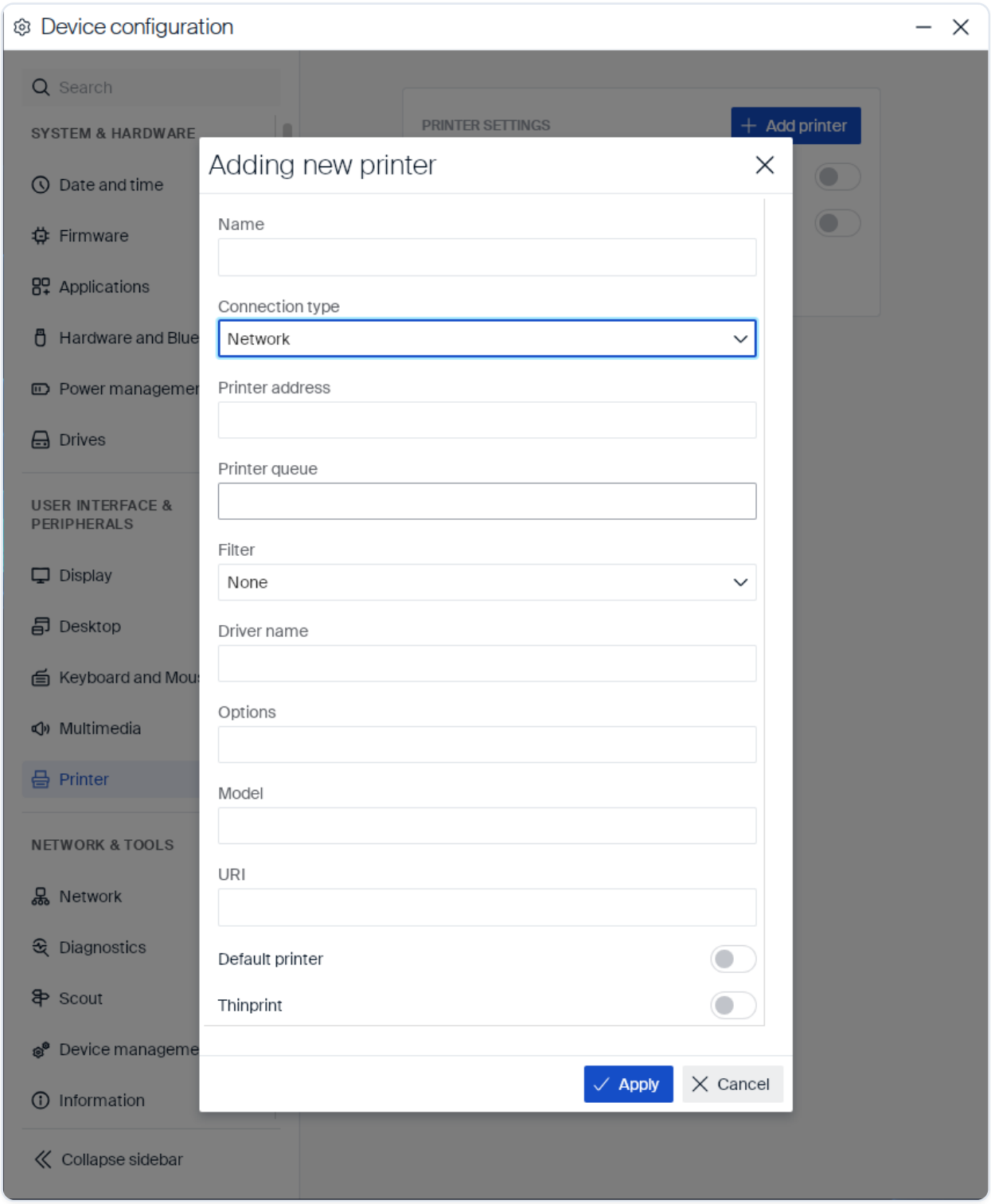


2. Fill in the following fields to identify the printer and how it connects to the device:

FIELD	TYPE	DESCRIPTION
Name	Text field	Name the printer appears under in application print dialogs.

FIELD	TYPE	DESCRIPTION
Connection type	Dropdown	`USB`, `Network`, `Parallel`, or `COM 1`.
Printer address	Text field	Hostname or IP address of the network printer. Only shown when **Connection type** is `Network`.
Printer queue	Text field	Print queue name on the network printer. Only shown when **Connection type** is `Network`.
Filter	Dropdown	`None`, `Text`, or `PCL2`.
Driver name	Text field	Identifier of the printer driver to use.
Options	Text field	Additional driver options.
Model	Text field	The printer's exact model, used to select a matching driver.
URI	Text field	Connection URI for the printer.
Default printer	Toggle	Makes this the default printer.
Thinprint	Toggle	Enables printing through a ThinPrint connection.
Class	Text field	ThinPrint class to use. Only shown when **Thinprint** is on.

Selecting `Network` as the **Connection type** shows the **Printer address** and **Printer queue** fields right below it:

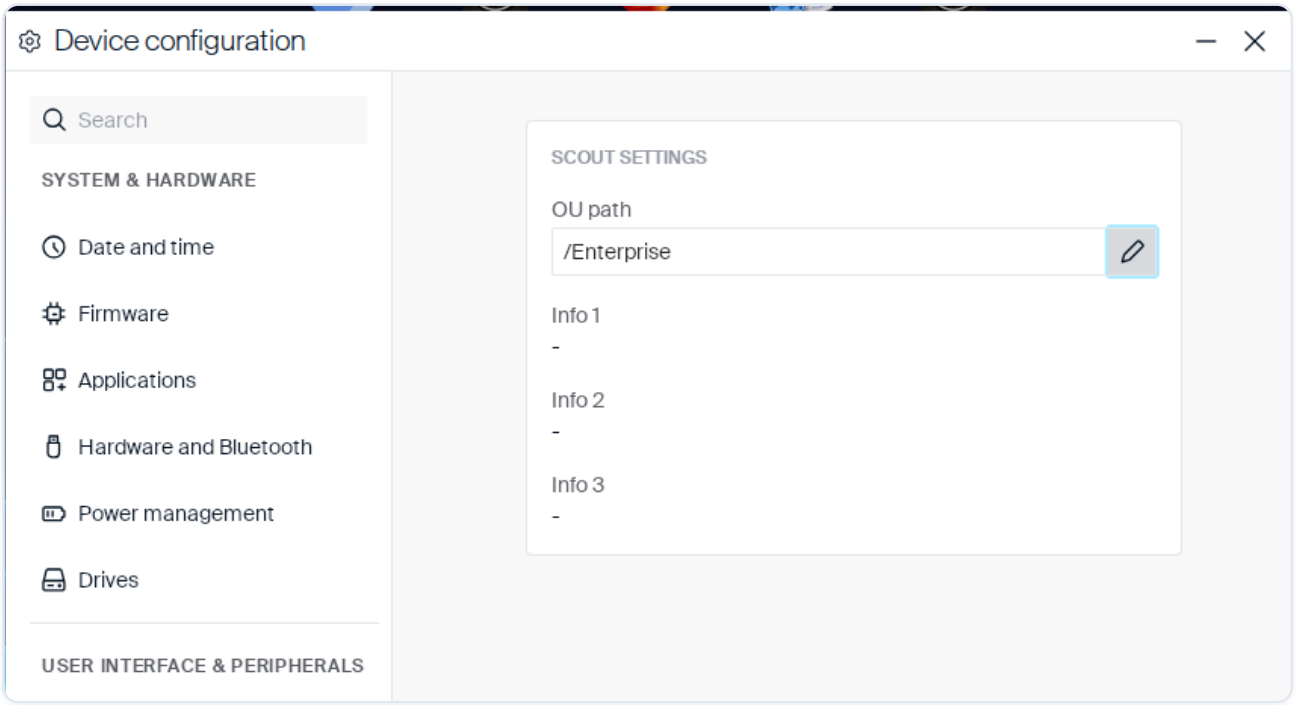


3. Click **Apply** to save the printer, or **Cancel** to close the dialog without saving it.

The printer appears in the printer list. To remove a printer, use its delete action in the list.

Scout

Scout shows the device's OU path (its position in the organizational unit hierarchy) and info fields, and lets you change the OU path if permitted by the Scout configuration.



These fields show where the device sits in Scout, and any extra information an administrator has attached to it:

SETTING	TYPE	DESCRIPTION
OU path	Text field	The organizational unit the device belongs to in Scout, for example `/Enterprise`. Click the pencil icon to edit it, if permitted by the Scout configuration.
Info 1, Info 2, Info 3	Info	Additional device information fields defined in Scout. Shown as `-` when not set.

Clicking the pencil icon next to **OU path** opens a **Change OU assignment** dialog with a dropdown to pick the new path, and **Ok**/**Cancel** buttons:

Appendix

eLux 7 partitions

IP ports

SNMP

eLux® 7 partitions

The partition layout of eLux 7 offers more space for applications on devices that have more than the minimum hardware requirements of 16 GB flash memory. The capacity of the system partition for applications is determined by the size of the flash memory. Many more applications can be used on a device with a flash size of 128 GB or 256 GB than on a device with a flash size of 16 GB.

The flash memory of a client is divided into four or five partitions during the eLux installation. Each partition is reserved for a dedicated purpose and is only called up when executing tasks related to the relevant partition. All partitions are recreated during a recovery installation.

> **Note:**
>
> You can find the minimum requirements in the article System requirements.

Partitions overview

PARTITION	PURPOSE	RECREATED WITH	OTHER
System	Reserved for the firmware (software packages)	Scout Update command	–
Boot	Boot section	–	–
Setup	Device configuration	Factory reset command	Does not affect the system partition with installed firmware
	Local application definitions		
Update	Software delivery in advance (before firmware update) via Scout command or	Scout Delivery command with the **Clean up update**	–

PARTITION	PURPOSE	RECREATED WITH	OTHER
	notification. Signature check for eLux software packages. Devices with an update partition can be used as a Dynamic Proxy (Provider) for firmware updates.	**partition before delivery** option	
Custom	Optionally usable by applications with separate storage space requirements.	–	Unencrypted partition without a specific mount point. No custom partition is created on devices with less than 16 GB of flash memory.

Network requirements

All contents relating to ports, protocols and network requirements are covered by the **network requirements** guide. For further information, see Network requirements guide.

SNMP

SNMP (Simple Network Management Protocol) is a network protocol for monitoring and controlling network devices.

For eLux® RP 6 and newer, SNMPv3 is used.

> **Note:**

> The SNMP environment package includes the full net-snmp client toolset, including **snmpget**, **snmpwalk** and **snmpset**, so you can query SNMP status information locally on the device without third-party software.

> Configuring SNMP

SNMPD and SNMP Configuration Directives

Configuring SNMP

1. In **Software** in UniconOS Management Board, add the **SNMP Environment** package to the image of the devices, and update the devices. For how to add a package to an image, see Changing an image.

   From UniconOS 2611 on, the package is in the AllPackages container archive. For earlier versions, download it from our myelux.com portal, under **eLux Software Packages**, for your eLux version, under **Add-On**. Then add it to the container, see Adding packages.

2. If there is no `/setup/snmp/snmpd.conf` on the devices, transfer the configuration file `snmpd.conf` to `/setup/snmp/snmpd.conf` on the devices by using the Scout feature Files configured for transfer.

   Or:

   Modify the `terminal.ini` file by using the Adding individual file entries feature of Scout. Example:

File	/setup/terminal.ini
Section	SNMPD
Entry	rocommunity
Value	secret

3. Optionally, define further SNMPD and SNMP Configuration Directives and modify the `terminal.ini` file under `SNMPD`. Examples:

   `syscontact=

   syslocation=testcenter doDebugging=1`

   For further information on SNMPD Configuration Directives, see http://www.net-snmp.org.

The `SNMPD` section of the `terminal.ini` file is evaluated by the client, and the file `/setup/snmp/snmpd.local.conf` is created. An existing `/setup/snmp/snmpd.conf` is overwritten.

If the configuration file does not exist, the file `/setup/snmp/snmpd.local.conf` is created with default values.

> **Note:**
>
> Restart the device after Scout delivers a new or changed SNMP configuration.

NOTES ON CONFIGURING SNMP V3

- When you define users (**createUser**), set a password with at least 8 characters.
- For the authentication method, define `authPriv` or `authNoPriv`.

> **Note:**
>
> For SNMP v2, you can use `noAuthNoPriv` as the authentication method.

SNMPD and SNMP Configuration Directives

The following table refers to the eLux® software package snmp. For further information on using SNMP with eLux, see SNMP.

For further information on SNMP commands, see http://www.net-snmp.org.

DIRECTIVE	SYNTAX
authtrapenable	1 | 2 (1 = enable, 2 = disable)
trapsink	host [community] [port]
trap2sink	host [community] [port]
informsink	host [community] [port]
trapsess	[snmpcmdargs] host
trapcommunity	community-string
agentuser	agentuser
agentgroup	groupid
agentaddress	SNMP bind address
syslocation	location

DIRECTIVE	SYNTAX
syscontact	contact-name
sysservices	NUMBER
interface	name type speed
com2sec	name source community
group	name v1|v2c|usm security
access	name context model level prefix read write notify
view	name type subtree [mask]
rwcommunity	community [default|hostname|network/bits] [oid]
rocommunity	community [default|hostname|network/bits] [oid]
rwuser	user [noauth|auth|priv] [oid]
rouser	user [noauth|auth|priv] [oid]
swap	min-avail
proc	process-name [max-num] [min-num]
procfix	process-name program [arguments...]
pass	miboid command
pass_persist	miboid program
disk	path [minspace | minpercent%]
load	max1 [max5] [max15]
exec	[miboid] name program arguments
sh	[miboid] name program-or-script arguments

DIRECTIVE	SYNTAX
execfix	exec-or-sh-name program [arguments...]
file	file [maxsize]
dlmod	module-name module-path
proxy	[snmpcmd args] host oid [remoteoid]
createUser	username (MD5|SHA) passphrase [DES] [passphrase]
master	specify 'agentx' for AgentX support
engineID	string
engineIDType	num
engineIDNic	string

SNMP CONFIGURATION DIRECTIVES

DIRECTIVE	SYNTAX
doDebugging	(1|0)
debugTokens	token[,token...]
logTimestamp	(1|yes|true|0|no|false)
mibdirs	[mib-dirs|+mib-dirs]
mibs	[mib-tokens|+mib-tokens]
mibfile	mibfile-to-read
showMibErrors	(1|yes|true|0|no|false)
strictCommentTerm	(1|yes|true|0|no|false)
mibAllowUnderline	(1|yes|true|0|no|false)
mibWarningLevel	integerValue

DIRECTIVE	SYNTAX
mibReplaceWithLatest	(1|yes|true|0|no|false)
printNumericEnums	(1|yes|true|0|no|false)
printNumericOids	(1|yes|true|0|no|false)
escapeQuotes	(1|yes|true|0|no|false)
dontBreakdownOids	(1|yes|true|0|no|false)
quickPrinting	(1|yes|true|0|no|false)
numericTimeticks	(1|yes|true|0|no|false)
suffixPrinting	integerValue
extendedIndex	(1|yes|true|0|no|false)
printHexText	(1|yes|true|0|no|false)
dumpPacket	(1|yes|true|0|no|false)
reverseEncodeBER	(1|yes|true|0|no|false)
defaultPort	integerValue
defCommunity	string
noTokenWarnings	(1|yes|true|0|no|false)
noRangeCheck	(1|yes|true|0|no|false)
defSecurityName	string
defContext	string
defPassphrase	string
defAuthPassphrase	string
defPrivPassphrase	string

DIRECTIVE	SYNTAX
defVersion	(1|2c|3)
defAuthType	(MD5|SHA)
defPrivType	(DES)
defSecurityLevel	(noAuthNoPriv|authNoPriv|authPriv)

Crash and error reporting

Crash and error reporting sends reports about crashes and some errors in UniconOS to a Sentry project that you operate. You see the failure and the steps that led to it, instead of collecting logs from each device afterwards.

The feature needs UniconOS 2609 or later. It is **off by default** and sends nothing until you configure it.

Configuration

Three settings control the feature. They are in the `[Global]` section of `/setup/terminal.ini`. In Scout and UniconOS Management Board, set them as advanced file entries, one entry per key. For how to define an entry, see Advanced file entries.

Reporting runs only when `SentryEnabled` is on and `SentryDsn` is set. `SentryEnvironment` is optional:

KEY	VALUES	DEFAULT	DESCRIPTION
`SentryEnabled`	`1` or `true`	`0`	Turns reporting on. Any other value leaves it off.
`SentryDsn`	DSN URL	Empty	Where reports are sent. Reporting stays off while this is empty.
`SentryEnvironment`	Free text	Empty	A label that groups your devices in Sentry, for example `production`.

For example:

```
[Global]

SentryEnabled=1

SentryDsn=https://<key>@<your-sentry-host>/<project-id>

SentryEnvironment=production
```

Restart the device after you change a value. Reporting does not start or stop on a running device.

To turn reporting off, set `SentryEnabled=0` and restart the device. Reports that are still held on the device are discarded.

GETTING A DSN

The DSN (Data Source Name) is the address of your Sentry project. Devices send their reports to it. To get it:

1. In your Sentry organization, create a project with the **Native** platform.
2. Open **Settings** > **Projects** > *your project* > **Client Keys (DSN)**, and copy the DSN.

> **Note:**

> A DSN contains a key for sending reports to your project and is stored in plain text on the device. Treat it as a shared secret. Remove it from diagnostic files that you collect with Device diagnostics before you share them.

What is sent

Reports come from the UniconOS desktop, settings, login, and Scout agent components. They cover crashes and also some errors that do not crash the component, for example when it cannot reach the settings service.

Each report states:

- Which component failed
- The last steps it took before the failure
- The firmware version

- Your environment label

Crashes also include a crash dump, which is a copy of part of the failed process's memory.

Reports do not include:

- User names
- The device host name
- Asset tags or serial numbers
- The names of your published applications

> **Note:**
>
> UniconOS removes the data listed above from reports. A crash dump or an error message can still contain other data that the process held at the time. Restrict access to your Sentry project accordingly.

UniconOS Management

UniconOS Management, formerly Scout Enterprise Management Suite® or Scout for short, is a lean and easy-to-use end-point management solution for devices running UniconOS (formerly eLux®).

> **Note:**
>
> This guide describes how to work with the Scout Console. The Scout Console is end of life. It is no longer part of the standard installation from UniconOS Management 2611 LTSR, and it is not available for UniconOS Management 2611. UniconOS Management 2609 is the last release that installs it. To manage your devices, use UniconOS Management Board, see the UniconOS Management Board guide. For details, see Deprecation.

Overview

The Scout Enterprise Management Suite®, or Scout for short, is a lean and easy-to use end-point management solution for devices running eLux. Administrators can easily manage, configure, upgrade, update, secure and monitor even large numbers of devices.

This manual serves as a reference for the Scout Console functionality and related tools. It includes features from version 15.3 onwards. Scout and eLux features that have been added since version 2101 are referred to via footnotes.

- Features of the Scout Enterprise Management Suite
- Communication between devices and Scout Server
- Installation
- Keyboard shortcuts

Communication between devices and Scout Server

> **Note:**

> The certificate-based management protocol ensures secure communication between Scout Server and devices. For further information, see Certificate-based management protocol in the **Installation** guide.

> During system start, the eLux devices connect to their Scout Server and verify if their configuration data is up-to-date. Updated data may concern device configuration, application definitions, files defined for transfer and advanced file entries. For further information about identifying and transferring updated configuration data, see Configuration method.

The communication between a device and the server may proceed in three ways:

- The device accesses the Scout Server. The Scout Server has no updated configuration data.
  - The device continues starting up with its configuration.
- The device accesses the Scout Server. The Scout Server reports new configuration data and transfers the data to the device.
  - The device restarts using the new configuration.
- The device cannot access the Scout Server due to network or other problems which result in a management timeout, see Advanced network settings.
  - The device continues starting with its configuration. Depending on the handshake settings, the device retries connecting to be able to synchronize the configuration data. For further information, see Optimizing with handshake.

Updated configuration data can relate to device configuration (setup), application definition, files configured for transfer and advanced file entries.

During operation of a user's device there is no data exchange between the Scout Server and device. During shutdown, the device reports its current status to the Scout Server (except for VPN connections).

Interface of the Scout Console

- Organizational structure
- Icons in the tree view
- Windows
- Status bar
- Display settings
- Changing the language
- Search for devices, OUs or applications
- Moving and copying elements
- Switching OU to top-level

Windows

Click **View > Windows** to show further windows next to the organizational structure:

WINDOW	DESCRIPTION
Properties	Properties of the selected application, OU or device
	For further information, see Properties.
Assets (only for devices)	Hardware information of the selected device
	Is shown as a tab of the **Properties** window
	For further information, see Hardware information.
Dynamic Device Groups	Shows the defined Dynamic Device Groups (list view)

WINDOW	DESCRIPTION
Independent configurations	Shows OUs and devices that do not use the parent device configuration (list view)
	For further information, see Blocking inheritance - independent device configuration
Compare configurations	Shows differences in the device configuration between devices or OUs
OU devices/applications	Devices or applications of an OU view without icons (list view)
	Double-clicking on a device shows the corresponding device in the tree view. This feature can be disabled, see below.
All devices	Shows all devices without icons (list view)
	The device data are only loaded from the Scout database when you click the **Refresh** button. This is meant to prevent unintentional loading of huge amounts of data.
	The context menu offers the same functions as in the tree structure. Several functions such as commands can be applied to multiple devices. To do so, select the devices by pressing CTRL or SHIFT
	Double-clicking on a device shows the corresponding device in the tree view. This feature can be disabled, see below.
	To search window content, use the **Search** field of the toolbar, type (/en-us/unicon-elux-scout/2611/admin_guides/scout/interface_/interface_search.html).

Sorting columns

Click a column header to sort rows.

Showing or Hiding properties

Click the button to define which properties you want to show. Alternatively, use the context menu.

Disabling the Double-click shows device feature

By default, double-clicking a device within a device list causes the tree view to show the corresponding device. This behavior can be disabled.

Define the following registry entries with value type DWORD:32 and value 1:

HKEY_CURRENT_USERSoftwareUniConScoutSettings

DisableDoubleClick_OUDevices_View

DisableDoubleClick_AllDevices_View

DisableDoubleClick_DCG_View

Properties

The properties of the selected application, OU or device are displayed.

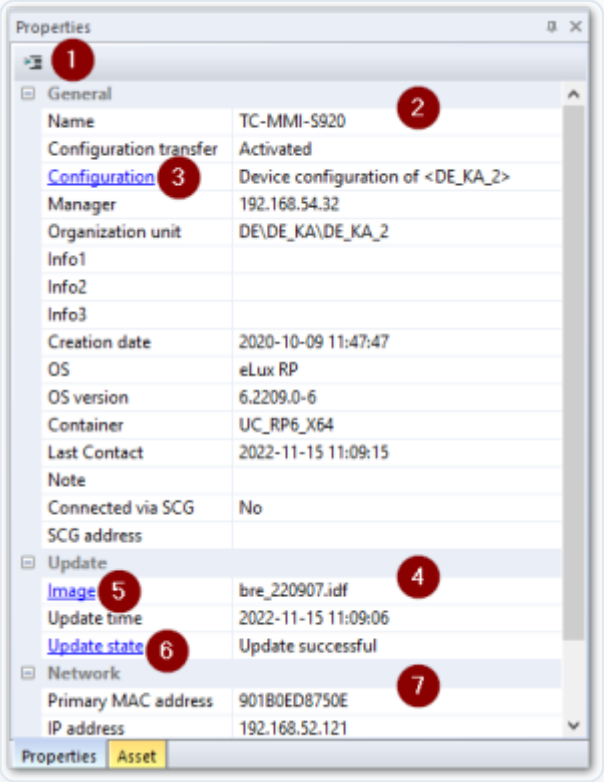
Many properties come from the device configuration (/en-us/unicon-elux-scout/2611/admin_guides/scout/setup/config_method.html).

Some device properties come from the Scout Server and are updated dynamically.

Date and time fields are shown according to the international standard ISO 8601.

> **Note:**

> By default, not all fields are displayed. To show or hide fields, click the icon



The figure shows the properties of a selected device.

1. Show, hide or move fields.
2. View general information on the device.
3. Click to open the device configuration the device is assigned to.
4. View information on the last firmware update.
5. Click to open the assigned image in ELIAS (provided **Options > ELIAS settings** is configured).
6. Click to open the update log.
7. View network information on the device.

DEVICE PROPERTIES

Examples:

OPTION	DESCRIPTION
Name	Host name of the device
Configuration	Origin of the device configuration, mostly inherited from parent instance
Manager	IP address of the Scout Server the device is assigned to

OPTION	DESCRIPTION
Info1-3	The **Info** fields are shown on the device in the **Configuration panel** under **Information** and can be enabled for users for editing (user rights). They are already provided in the First Configuration Wizard.
OS version	Version of client operating system
Container	eLux® container configured for the device in the firmware device configuration
Update time	Time stamp of the last firmware update
\^	For further information, see Command results and update information per device
Last contact	Time stamp of the last contact between server and device
\^	The field is not only updated on a device restart but also on each successful connection set-up from the server to the device.
Status	Example: Switched on
\^	The status of a device is triggered by the 'keep alive' mechanism and by active status messages of the device on start, shutdown, logoff, performing updates and more.
Status time	Time stamp of the last status refresh
Connected via SCG[1] [from Scout 15 2209 and eLux RP 6 2209]	Indicates whether a device is connected via [Scout Cloud Gateway]
SCG address[2] [from Scout 15 2209 and eLux RP 6 2209]	For SCG connections: Shows the FQDN and IP address of the Scout Cloud Gateway
Primary MAC address	Device address of the hardware (MAC=Media Access Control)

OPTION	DESCRIPTION
Simple device identifier	Temporary device identifier users may request for support cases
Client identifier	Globally unique identifier for a device or eLux Portable USB stick
Public address	If a device is connected via VPN, its public IP address is displayed next to its actual **IP address**. You can filter by the public IP address in the Scout Report Generator.
Partitions	The system, setup and update partitions are displayed with their respective sizes.
Screen (Screen info)	Shows the configured resolution and frequency of all connected monitors
\^	For multiple monitor layouts, additionally the output port and defined properties such as primary screen, rotation and the monitor's position (row/column) are displayed.[3] [from Scout 15 2204] If a monitor is not active, this is shown as a dedicated property.[4[[ ]from Scout 15 2209]] The values for row and column are then shown as [-1].

OU PROPERTIES

Examples:

OPTION	DESCRIPTION
OU	Shows the ID of an OU. In addition to the decimal value, you can show the hexadecimal value. This requires a new registry entry. Key: [HKEY_CURRENT_USERSoftwareUniConScoutSettings] ,Value name: DisplayHexOUID, Value type: DWORD:32, Value data: 1
Device count	Number of devices in the OU and the subordinate OUs

QUICK LINKS IN THE PROPERTIES WINDOW OF DEVICES AND OUS

- Make use of the links shown in blue to quickly browse the relevant configuration and information in each context. Double-click the links:

SELECTED ELEMENT	OPTION	DESCRIPTION
Device	**Configuration**	Opens the relevant **Device configuration**
Device	**Image**	Opens ELIAS with the image configured for this device in the relevant container
\^	\^	The connection to ELIAS is made with the data configured in the ELIAS settings of the [Scout Console]. For further information, see Access to applied images
Device	**Update State**	Double-click or [...] opens the [Update-Info] for the device providing information on performed updates. For further information, see Command results and update information per device
OU	**Configuration**	Opens the relevant **Device configuration**
OU	**Update**	Opens the relevant **Update** settings in the **Advanced device configuration** or **Advanced options**.
OU	**Screen, Printer, Mouse/ Keyboard**	Opens the relevant configuration (**Device configuration or Advanced device configuration**) for

SELECTED ELEMENT	OPTION	DESCRIPTION
		Screen, Printer or Mouse/ Keyboard

APPLICATION PROPERTIES

The details of an application depend on its type (/en-us/unicon-elux-scout/2611/admin_guides/scout/app_definition/app_general/app_add.html).

Hardware information

For the selected device, in the **Properties** window, on the **Assets** tab, device-specific hardware information is shown:

- Hardware information of the device
- Hardware information of connected devices such as USB devices and monitors

> **Note:**
>
> By default, not all fields are displayed. To show or hide fields, click the icon.
>
>

1. Information on the managed client device
2. Information on connected devices
3. Show, hide or move fields

HARDWARE INFORMATION OF A DEVICE

Examples:

OPTION	DESCRIPTION
Device type	Product details provided by the hardware manufacturer (character string)
Main memory	Main memory size
Flash memory	Flash memory type

OPTION	DESCRIPTION
Flash size	Size of the flash memory

HARDWARE INFORMATION OF CONNECTED DEVICES

Examples for a USB device:

OPTION	DESCRIPTION
Product Type	Type of USB device
Vendor	Vendor name
Serial number	Serial number of the USB device
Revision number	Firmware version of the USB device
Port number	Number of the port the USB device is connected to: bus number>-port number>

Moving and copying elements

Devices, OUs and applications can be moved from one OU to another OU within the tree view of the organizational structure. If inheritance is active, after you have moved a device or OU, it receives the properties of the new parent OU.

Moving devices, OUs or applications

1. In the tree view, show the source and target position of the relevant element.

   The target position can be the icon of the target OU or any valid position subordinate to the target OU.

2. Use a drag-and-drop operation to move the element from the source to the target position. or, Move the element via context menu or CTRL-X to the Clipboard and paste it via context menu or CTRL-V at the target position.
3. Confirm with **Yes**.

Confirm your changes again and the element is moved to the target OU.

Moving devices to another OU via context menu

1. In the tree view, in the **OU devices** window or in the **All devices** window, for the relevant device, open the context menu.
2. Click **Edit > Move...**
3. In the dialog, expand the organizational structure and select the target OU.
4. Confirm with **OK..**

Confirm your changes again and the element is moved to the target OU.

Copying applications

> **Note:**
>
> Applications in the tree view are application definitions and do not include software. The software must be configured and provided separately via IDF.

1. Show the source and target position of the relevant application in the tree view.

   The target position can be the icon of the target OU or the **Applcations** node subordinate to the target OU.
2. Use a drag-and-drop operation while pressing CTRL to move the application from the source to the target position.

   or,

   Copy the application via context menu or CTRL-C to the Clipboard and paste it via context menu or CTRL-V at the target position.
3. Confirm with **Yes**.

The application is copied to the target OU.

> **Note:**
>
> Applications can also be copied from any client device to a Scout OU. For further information, see Uploading applications from a device to Scout .

Device management

To be able to manage client devices with eLux or other operating systems, Scout must know their MAC address. There are several approaches to register new devices such as

- Self-registration of devices
- Searching for devices (Discovery)

New devices must be assigned to an organizational unit (OU). You can configure whether new devices

- are added to a specified OU (**default OU**)
- are assigned automatically by the OU filter according to definable criteria
- are created in terms of proxy profiles even before connecting Reserving device profiles

The way you want to deal with new devices is mainly defined under **Options > Advanced Options >** Devices.

> **Note:**
>
> As the devices are organized hierarchically in OUs, you can use Dynamic Device Groups to apply commands to several devices irrespective of their OUs.

- Self-registration of devices
- DHCP configuration
- Searching for devices (Discovery)
- Reserving device profiles
- Device names
- MAC address
- Secure device management with Scout
- OU filter
- Dynamic Device Groups
- Device relocation between Scout Servers

Self-registration of devices

By default, the first time a device boots, it automatically searches for an available Scout Server. The device requires the IP address of the Scout Server.

Requirements for self-registration:

- The device must be in initial state (either upon delivery or by performing a factory reset)
- The device must be connected to the network
- The Scout IP address must be configured in one of the following ways:
- DHCP: A configured DHCP option is set to the IP address/name of the Scout Server. You can also specify more than one Scout Server and a destination OU. For further information, see DHCP configuration.

or

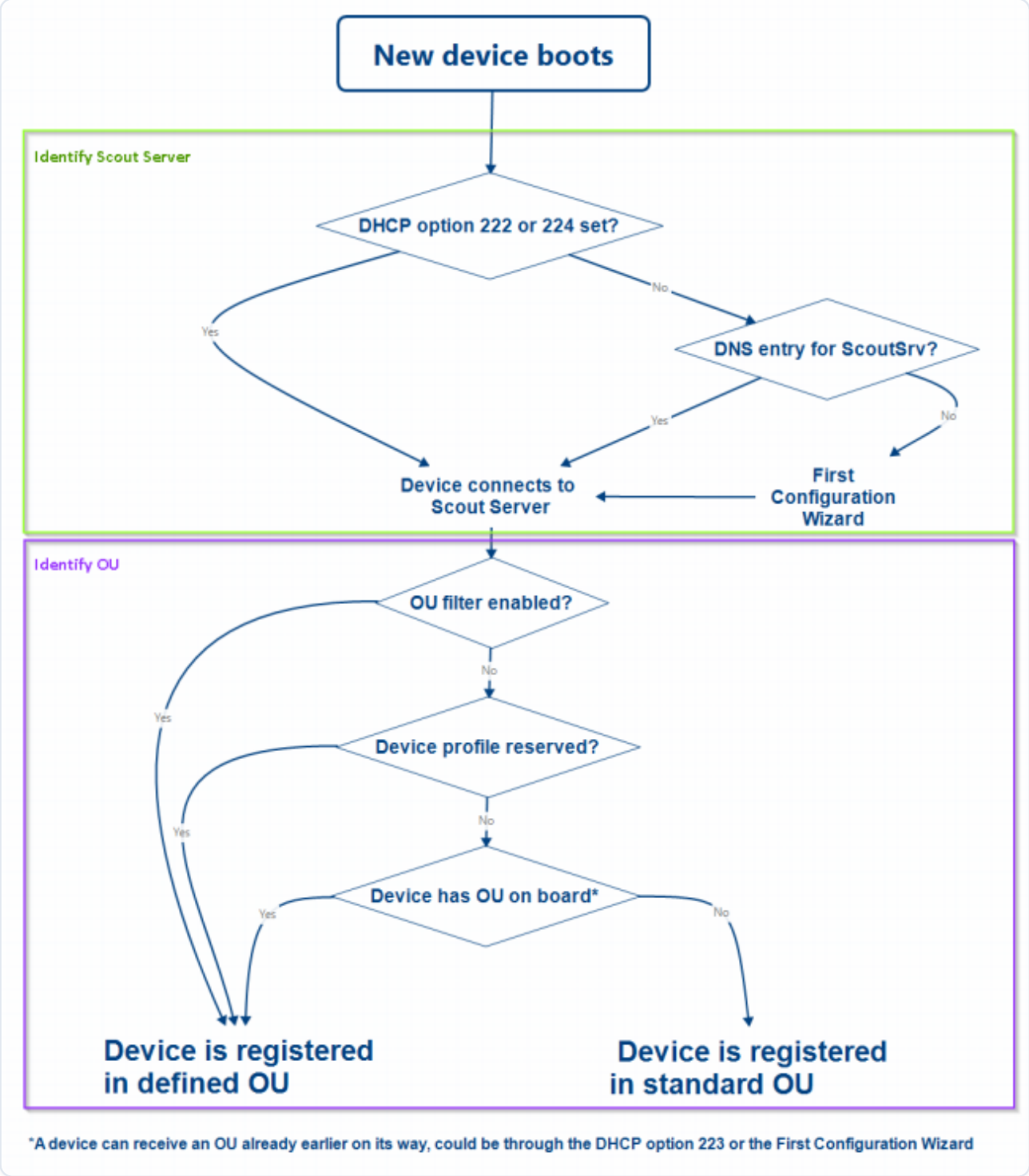
- DNS: The DNS server resolves the host name ScoutSrv (no case-sensitivity).

If the Scout Server's IP address cannot be determined, neither by the DNS nor by DHCP, a First Configuration Wizard automatically runs on the device to help the local user through the initial configuration.

Registering a device automatically

- Turn the device on.

If the requirements for self-registration are met, the device contacts the Scout Server and enters itself in the defined OU or the standard OU. It receives the configuration of the OU and is restarted with the new configuration.



The flow chart roughly shows the way a new device is assigned to a Scout Server and to an OU. Details such as the **Accept only known devices** have not been considered.

Searching for devices (Discovery)

Based on the IP address, you can search for devices throughout the entire network or within specific subnets. Any matching devices are automatically registered to Scout and are added to the specified OU **(Destination group)**. The devices are restarted and receive the configuration of the

destination group (device configuration, application definitions, files defined for transfer and advanced file entries).

> **Note:**
>
> If the OU filter is active, the filter specifies the destination group or groups. For further information, see Devices.

> Requirements:

- The devices are turned on and connected to the network.
- The devices are provided with valid IP addresses.
- The device password is known.

Searching and registering devices

1. Make sure that your destination OU is configured correctly.
2. Select **Options > Search devices**.

3. Edit the following fields:
   a. Enter the first IP address of the range.
   b. Enter the number of IP addresses within the range (restricted to 255).
   c. Enter the last IP address of the range.
   d. Enter the device password (default: **elux®**).
      - The password must match the password currently set on the individual devices.
   e. Select the OU you want to assign the devices to.
      - Default is the predefined **Lost&Found** group with the base device configuration.

> **Important:**
>
> If the field is disabled, the OU filter is active and the matching devices are assigned according to the OU filter rules.

a. The user is informed by a message about the upcoming device restart.

   - Specify in seconds how long the message shall be displayed. If the time period is set to , the system will wait until the user confirms.

b. Allows the user to suppress the device restart. The configuration is not updated until the device is restarted.

4. Confirm with **OK..**

The matching devices receive the IP address of the managing . The devices are assigned to their destination group and are restarted. The devices inherit the configuration of their new OU. Local non-protected device configuration is overridden. With immediate effect, on each restart, the devices connect to their and, if available, are given the latest configuration and application definition data.

If a device profile has been reserved for a device, the predefined profile is automatically assigned at Discovery.

To modify response time and maximum search time for the Discovery feature, choose **Options > Advanced options > Devices > Discover devices**.

> **Note:**

> Devices already registered to Scout are not modified, but their status is updated when connected.

MAC address

The MAC address is the hardware address or physical address of a network device. Scout used to identify devices by their MAC addresses. In the meantime, the **client identifier** does this job.

MAC addresses can be used for the following tasks, for example:

- Import devices
- Reserve device profiles
- Create dynamic device groups

In Scout, MAC addresses are normally represented as 12-digit numbers without separators. Example: 901B0E01CE84

From , you can also use colons or hyphens as separators between the 6 pairs:

12-digit MAC address without separators	001122334455
12-digit MAC address with colons	00:11:22:33:44:55
12-digit MAC address with hyphens	00-11-22-33-44-55

Note:

The MAC address usually is determined from the network interface (Lan or WLAN card).

Secure device management with Scout

An enhanced security level is available for adding new devices to .

Clients that are registered with their MAC address in the database Reserving device profiles are accepted by the Scout Server and can be integrated into the Scout management. In contrast, devices having an unknown MAC address are not accepted and therefore cannot be managed by Scout.

Accepting known devices only

1. In the , select **Options > Advanced options > Devices > New**.
2. Select the **Accept only known devices** option.

If an unknown device tries to contact the Scout Server, an error message is displayed on the device stating that a connection to the Scout Server was denied.

Note:

Only the requests of those devices are accepted whose MAC addresses have been saved to the Scout database by a device import or device profile.

OU filter

OU filters can be used for automatic assignment of devices to an organization unit (OU). The assignment is based on predefined criteria. This is particularly helpful when registering new devices or relocating existing ones.

There are two options for configuring an OU filter:

- The **subnet OU filter** uses the devices' network address for filtering
- The **user-defined OU filter** uses configured asset information of the devices for filtering

You can only use one filter at a time. For each filter, you can define multiple filter rules and specify the sequence you want the rules to be applied in.

Once defined, the filter rules are retained unless deleted manually. Deactivate the filter rules that are currently not required but that you want to keep for future use.

The OU filter has precedence over

- OU assignment of devices via the DHCP option 223
- Discovery of new devices via Scout
- selecting the OU in the First Configuration Wizard locally on the device
- the default OU specified in **Advanced options > Devices**.

For individual devices, you can ignore the OU filter **Advanced device configuration > Management**.

> **Note:**

> OU filters are included when you export the data category **Advanced options**. For further information, see Import/Export.

- Setting up an OU filter as subnet filter
- Setting up an OU filter as user-defined filter
- Editing OU filter rules
- Deactivating OU filter for individual devices
- Re-activating OU filter

Setting up an OU filter as subnet filter

You can use the OU filter to filter on client network addresses and assign the matching devices to an OU.

1. Click **Options > Advanced options > Devices**.
2. Under **New devices**, select the **Assign OU depending on OU filter** option. If required, click ••• to open the **OU assignment** dialog.
3. Under **Filter type**, select Subnet filter (device network address).
4. Create a new filter rule. Edit the following fields:

OPTION	DESCRIPTION
Network address	Enter the scope of IP addresses. Example: 192.168.16.0 covers all IPs starting with 192.168.16.
Netmask	Enter the relevant network prefix to define relevant devices.
Destination OU	To browse the OU list, click and select the OU you want the specified devices to go.

5. Click **Add**.

   The filter rule is displayed in the field below.

6. If required, add more filter rules and configure them. For further information, see Options for OU filter rules.
7. In the bottom section, from the list **Non-matching devices**, select where you want to keep the non-matching devices.

   > **Important:**
   >
   > If you select , all non-matching devices and even devices already assigned to other OUs are reassigned to the default OU.

8. Review all active filter rules thoroughly to avoid unintentional assignments.

9. Confirm with **OK**..

All active filter rules are processed. On the next restart, the matching devices are assigned to the OUs as defined by the OU subnet filter. Any additional user-defined filter rules will not be taken into account.

OU filter as a user-defined filter

You can filter by configured asset information of the devices to assign the matching devices to the appropriate OUs.

FILTER RULES WITH OU FILTER TEXT

eLux® devices send an **OU filter text** field containing their device information to the Scout Server. You can use the **OU filter text** field in the Scout Report Generator and for the user-defined OU filter. It includes the following information:

Host name, OS name, OS version, serial number, supplier, device type, UEFI/BIOS, CPU speed, model, kernel version, flash type, flash size, RAM size, graphics, IDF name, MAC address.

- A filter rule can contain one or more filter criteria.
- A filter criterion consists of three parts:

  asset information string from `OU filter text` | logical operator = | value you want to filter by

Asset information elements:

ELUX_HOSTNAME=Alrin;

ELUX_OSNAME=eLux;

ELUX_OSVERSION=7.2507.0-3;

ELUX_SERIAL=7250O422542;

ELUX_SUPPLIER=WYSE;

ELUX_DEVICETYPE=ZQ Class;

ELUX_BIOS=V4.6.5.4 R1.15.0;

ELUX_CPU=3999;

ELUX_PRODUCT=ZQ Class;

ELUX_KERNEL=6.12.30unicon0;

ELUX_FLASH=16GB SATA Flash;

ELUX_FLASHSIZE=15272;

ELUX_MEMORY=8192;

ELUX_GRAPHICS=ATI AMD Radeon HD8330E;

ELUX_IDF=recovery.idf;

ELUX_MAC=7CD30A22D0AE

ELUX_IP=192.168.10.21

ELUX_NETMASK=255.255.255.0

ELUX_BROADCAST=192.168.10.255

ELUX_NETADDR=192.168.10.0

ELUX_NETCIDR=24

ELUX_DOMAIN=crop.company.com

- Use the logical operators **AND** and **OR** to link multiple filter criteria. Use capital letters for the operators.
- Wildcards are not supported, but all matches will be found that begin with the specified string.

Example filter rules:

`[ELUX_OSNAME=eLux AND ELUX_OSVERSION=7.2507.0-3]` `[ELUX_DEVICETYPE=D3314-A1 OR ELUX_DEVICETYPE=ZQ Class]`

SETTING UP A USER-DEFINED OU FILTER

1. Click **Options > Advanced options > Devices**.
2. Under **New devices**, select the **Assign OU depending on OU filter** option. If required, click ••• to open the **OU assignment** dialog.
3. Under **Filter type**, select User-defined filter (configured asset information).
4. Create a new filter rule.

OPTION	DESCRIPTION
Filter rule	Create a filter rule consisting of one or more filter criteria, see above.
Destination OU	To browse the OU list, click ••• and select the OU you want the specified devices to go.

5. Click **Add**.

   The filter rule is displayed in the field below.

6. If required, add more filter rules and configure them. For further information, see Options for OU filter rules.
7. From the list **Non-matching devices**, select where you want to keep the non-matching devices.

   > **Important:**
   >
   > If you select , all non-matching devices and even devices already assigned to other OUs are reassigned to the default OU.

8. Review all active filter rules thoroughly to avoid unintentional assignments.
9. Confirm with **OK**..

All active filter rules are processed in the specified order. On the next restart the matching devices are assigned to the OUs as defined by the OU user-defined filter. Any additionally defined subnet filter rules will not be taken into account.

BEST PRACTICES

The asset information **ELUX_IP** corresponds to the **(Peer) IP address**.

When eLux endpoints connect through the Scout Cloud Gateway (SCG), the peer IP address reflects the **SCG-assigned VPN IP** (e.g. `10.100.100.0` default range).

When devices connect directly to a **Scout Server**, the peer IP equals the endpoint's **local IP address**

> **Notes**

> The Scout IP address always reflects the endpoint's local IP.

Example OU filter expression
If you want to assign eLux endpoints based on local network ranges (instead of SCG assigned VPN IPs), you can use a filter like this:

```
ELUX_NETADDR=192.168.168.0 AND ELUX_NETMASK=255.255.255.0
```

This will ensure OU assignments reflect the device's **true local subnet**, instead of the SCG VPN IP range

> **Important**

> If you are defining more complex and nested expressions by using brackets, these brackets - as control characters - need to be escaped.

> Example with brackets and escaping:

```
\(ELUX_CPU=512 AND ELUX_MEMORY=1024\) OR \(ELUX_OSNAME=eLux AND ELUX_HOSTNAME=elux7-vm\)
```

Options for OU filter rules

Once you have defined OU filter rules they remain until they are deleted explicitly. You can edit the filter rules in the following ways:

OPTION	ACTION	DESCRIPTION
Create new filter rule	Click **Add**	User-defined filter: The filter criteria of the Filter rule field and the selected Destination OU are added as a new filter rule to the list. Syntax of a filter criterion: String from OU filter text=value, You can combine two or more filter criteria by using one of the

OPTION	ACTION	DESCRIPTION
		logical operators AND or OR. Use capital letters for the operators. For examples, see OU filter as a user-defined filter. Subnet filter: The data from the fields Network address, Netmask and OU are added as a new filter rule to the list.
Delete a filter rule	Select filter rule and click [Delete]	The selected filter rule is deleted.
Edit filter rule	Select filter rule and press [F2] or double-click	Modify the elements of a filter rule right in the list.
Activate / deactivate a filter rule	Select/Clear [Activated] option	Deactivated filter rules are not executed. Newly added filter rules are active by default.
Change sequence of processing (user-defined filter)	Edit [Sequence] field	Filter rules with low sequence number are processed prior to filter rules with high sequence number.

1. Review all active filter rules thoroughly to avoid unintentional assignments.
2. For the entire filter, under **Non-matching devices**, select where you want to keep the non-matching devices.

> **Important:**
>
> If you select assigned to the default OU, all non-matching devices and even devices already assigned to other OUs are reassigned to the default OU.

Deactivating OU filter for individual devices

If the OU filter is enabled, all active filter rules are applied and the matching devices are assigned to the specified OU on their next restart. To exclude an individual device from the filter, deactivate the OU filter for that device.

1. For the relevant device, open **Advanced device configuration > Management**.
2. Under **New devices**, select **Ignore OU filter**.
3. Confirm with **OK**..

Or:

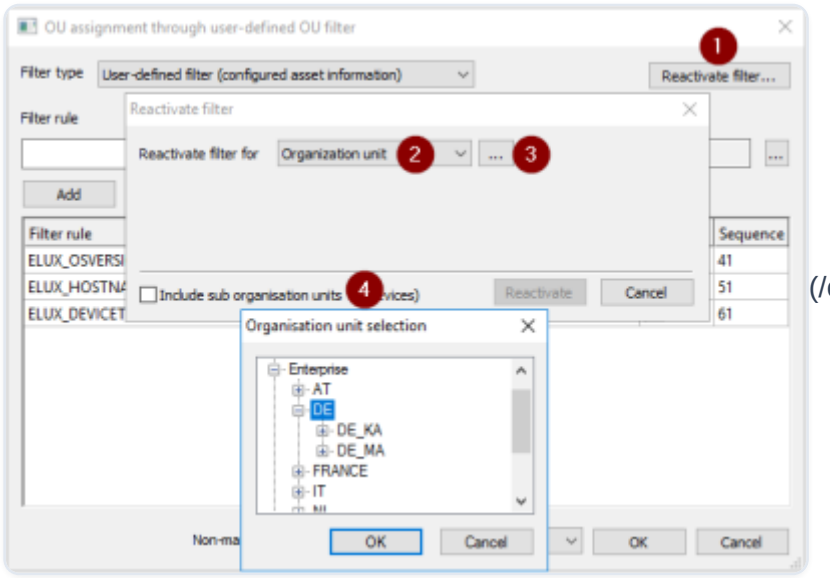
1. By using a drag-and-drop operation, move the device to another OU.
2. Confirm with **OK**..

The device is assigned to the new OU and the OU filter is deactivated for this device.

Re-activating OU filter

The OU filter can be deactivated for individual devices - either by applying the relevant option in the Scout Console or by moving devices by a drag-and-drop operation. To bring the relevant devices back to the OU filter, use the Scout Console option **Re-activate filter**.

1. Click **Options > Advanced options > Devices**.
2. Under **New devices**, select the **Assign OU depending on OU filter** option. If required, click ••• to open the **OU assignment** dialog.
3. In the upper right section, click **Reactivate filter...**(1).



(/en-us/unicon-elux-scout/2611/media/admin_guides/scout/ou_filter_reactivate.png)

4. Under **Reactivate filter for** (2), select the scope of devices for which you want to re-activate the OU filter: for all devices, for an individual OU or for Dynamic Device Groups.
5. To re-activate the OU filter for OUs or Dynamic Device Groups, click ••• (3) next to the list-field and select an OU or Dynamic Device Group.

   To include devices of subordinate OUs, select (4).
6. Confirm with **Reactivate** and **OK**.

The OU filter is re-activated for the selected devices.

Dynamic Device Groups

Dynamic Device Groups enable administrators to run cross-OU commands for freely definable device groups. For example, you can send a message to all devices with a specific image throughout the whole organization. Or you can run a UEFI/BIOS update on all devices with a specific UEFI/BIOS version, across all OUs. Even device relocation to another Scout Server can be applied to a Dynamic Client Group. The following features can be applied:

- Commands
- Configuration run
- Notifications for software deliveries or firmware updates
- Notifications for relocation

Dynamic Device Groups are based on reports created in the Scout Report Generator. The reports are exported to the Scout Console once, and from that point onward, are displayed as a **Dynamic Client Group**. Commands applicable to OUs or to individual devices can also be applied to Dynamic Device Groups.

> **Note:**

> The report layout must include the Client Identifier field. The report type must be a list of **Devices** or **Assets**. For further information on defining , see Creating Dynamic Device Groups in the Scout Report Generator guide.

> Dynamic Device Groups are displayed in the Scout Console in a special window and remain there for re-use until they are deleted. They can be updated any-time with a click.

User rights

When you create Dynamic Device Groups, the user access rights are respected as defined in the administrator management.

By default, executing commands and other functions on a Dynamic Device Group is subject to the object rights configured for the relevant devices and OUs. This can result in a function not being executed because at least one device does not have the required object rights. To avoid this, disable the check for the underlying object rights under **Security > Manage administrators > DCG rights**.

- Using Dynamic Device Groups
- Special form of Dynamic Device Groups by import

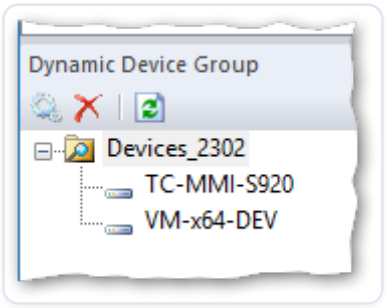
Using Dynamic Device Groups

Dynamic Device Groups are usually based on reports created in the Scout Report Generator and exported to the Scout Console.

For further information on defining and exporting DCGs, see Creating Dynamic Device Groups in the Scout Report Generator guide.

DISPLAYING DYNAMIC DEVICE GROUPS

- In the Scout Console, click **View > Window > Dynamic Device Groups**.



The **Dynamic Device Groups** window is displayed. The Dynamic Device Groups can be expanded to show the matching devices.

> **Note:**
>
> The Dynamic Client Group shows those devices that have matched the criteria at the time of the latest report generation. Make sure that the Dynamic Client Group is up-to-date.

> For a selected Dynamic Client Group, the **Properties** window shows the **Creation date, Number of devices and Filter** criteria of the used report. The creation date refers to the date of the latest generation of the report the Dynamic Client Group is based on, and thus indicates if the Dynamic Client Group is up-to-date.

If, for example, new devices have been integrated into the database and match the criteria of the report, the Dynamic Client Group is not up-to-date any longer. You can, however, update the Dynamic Client Group by re-creating the report right from the Scout Console.

If a Dynamic Client Group is not needed any longer, use the button to delete it. The report the Dynamic Client Group was based on remains unaffected.

UPDATING DYNAMIC DEVICE GROUPS

1. In the Dynamic Device Groups window, select the relevant device group.
2. On the toolbar of the **Dynamic Device Groups** window, click the **Recreate** button .

The relevant report is re-created and exported. The resulting devices are shown below the Dynamic Client Group as extracted from the database. In the **Properties** window, in the **Creation date** field, the current point of time is displayed.

> **Note:**

> The **Refresh** button refers to the view only. The report is not updated by this command.

APPLYING COMMANDS AND NOTIFICATIONS TO DYNAMIC DEVICE GROUPS

1. In the **Dynamic Device Groups** window, select the relevant Dynamic Client Group, and then check the information shown in the **Properties** window.
2. To update the Dynamic Client Group, click the **Re-create** button. This way you ensure that exactly the currently matching devices are affected.
3. Open the context menu of the Dynamic Client Group and select a command or notification.

If the required object rights are available, the commands and notifications are applied to the matching devices, irrespective of their OU. The available commands can also be scheduled for later execution.

Special form of Dynamic Device Groups by import

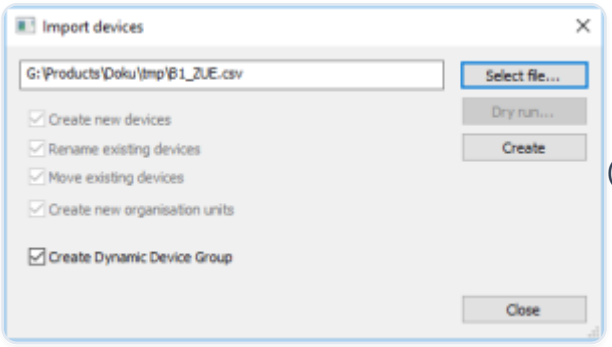
To create a Dynamic Device Group, instead of using the Scout Report Generator, you can use the Import feature of the Scout Console and make up a device group based on a device list containing MAC addresses. The advantage is that you are completely free to select any devices you want. They only have to be registered in Scout. Note that the Import feature here is not used to import devices.

Creating Dynamic Device Groups by import

REQUIRES
The relevant devices must be listed with their MAC addresses in a .csv file. Each line must begin with a MAC address. The lines may contain further information but only the MAC address is evaluated.

1. In the Scout Console, click **File > Import > Devices**.
2. In the **Import devices** dialog, in the bottom section, select the **Create Dynamic Device Group** option.

   All options related to the device import are disabled.

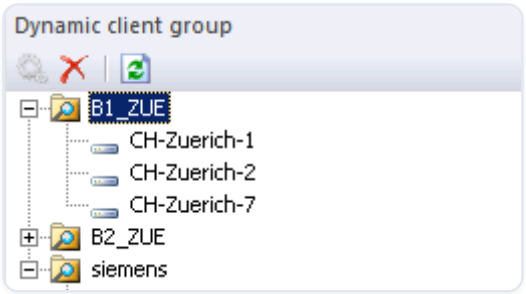
   

   (/en-us/unicon-elux-scout/2611/media/admin_guides/scout/dia_import_dcg.png)
3. Click **Select file...** and select the relevant .csv file from the file system.
4. Click **Create**.

The .csv file is evaluated. Scout creates a new Dynamic Client Group. It contains all devices of the csv list whose MAC address is registered in Scout. The dynamic device group adopts the name of the .csv file.

DISPLAYING DYNAMIC DEVICE GROUPS

- In the Scout Console, click **View > Window > Dynamic Device Groups**.



The **Dynamic Device Groups** window is displayed. The Dynamic Device Groups can be expanded to show the matching devices.

> **Note:**

> Dynamic Device Groups that have been created via the import feature cannot be updated with the **Re-create** button. To update the Dynamic Client Group, after having modified the device list, save the *.csv file under the same name and perform a new import as described above.

> For a selected Dynamic Client Group, the **Properties** window shows some information such as the **Creation date** and **Number of devices**. The **Filter** field shows the entry created by device import.

APPLYING COMMANDS AND NOTIFICATIONS TO DYNAMIC DEVICE GROUPS

1. In the Dynamic Device Groups window, select the relevant Dynamic Client Group. Check the information shown in the **Properties** window.
2. Open the context menu of the Dynamic Client Group and select a command or notification.

Commands and notifications are applied to the matching devices, irrespective of their OU. The available commands can also be scheduled for later execution.

Device relocation between s

Relocating devices from one Scout Server to another can be very helpful in different scenarios of device migration, for example when relocating devices from a test/QA server to a production server, or consolidating several Scout Servers to a single server (server fusion).

Client relocation can be performed with and without the devices verifying the availability of the target server. A so-called **offline** relocation does not require the target server to be physically available at the time of relocation. Example: External suppliers, in their environment, set up devices to be used in the customer's environment.

Local configuration can either be included in the transfer or left on the source server.

- Relocation procedure
- Relocation procedure / offline
- Relocation flow chart
- Initiating device relocation
- Scheduling the relocation process
- Deleting relocation notification

Relocation procedure

The relocation procedure is initiated by the source server (device-releasing server) and, by default, completed by the target server (device-receiving server). The actual relocation procedure, however, is performed by the devices. They check the conditions.

Relocation for the relevant devices is triggered by a notification named **Initiate device relocation** in the of the source server. The notification includes all required details. On the next device restart, the devices receive the new configuration data of the source server and evaluate the relocation notification.

The devices then check whether the transmitted target server's address is available via the network. Then, the relocation process will be started and the devices will be deleted from the source server.

The new devices are assigned to the specified OU on the target server. If no destination OU has been specified, the default OU or the OU defined by OU filter rules is used (configured in **Advanced options > Devices > New devices**).

The relocation procedure is completed by an automatic device restart which activates the configuration of the target server. If configured, locally defined configuration is retained. If the OU filter is used, an additional device restart is provoked by the system after assignment.

OPTIONS

In contrast to the default procedure, the administrator can configure the following options in the notification:

- Offline relocation

The devices are removed from the source server without checking whether the target server is available. They move to the target server only when they can connect to it.

- Retain local configuration on new Scout Server1

If on the source server is configured that users' local configurations in unlocked fields should not be overwritten by server-side configurations (), then the local configurations are carried along. The relocated receive the target server's device configuration only where there are no local configurations.

> **Important:**

> Do `not` reserve device profiles by entering MAC addresses of the new devices on the target server before device relocation.

Relocation procedure / offline

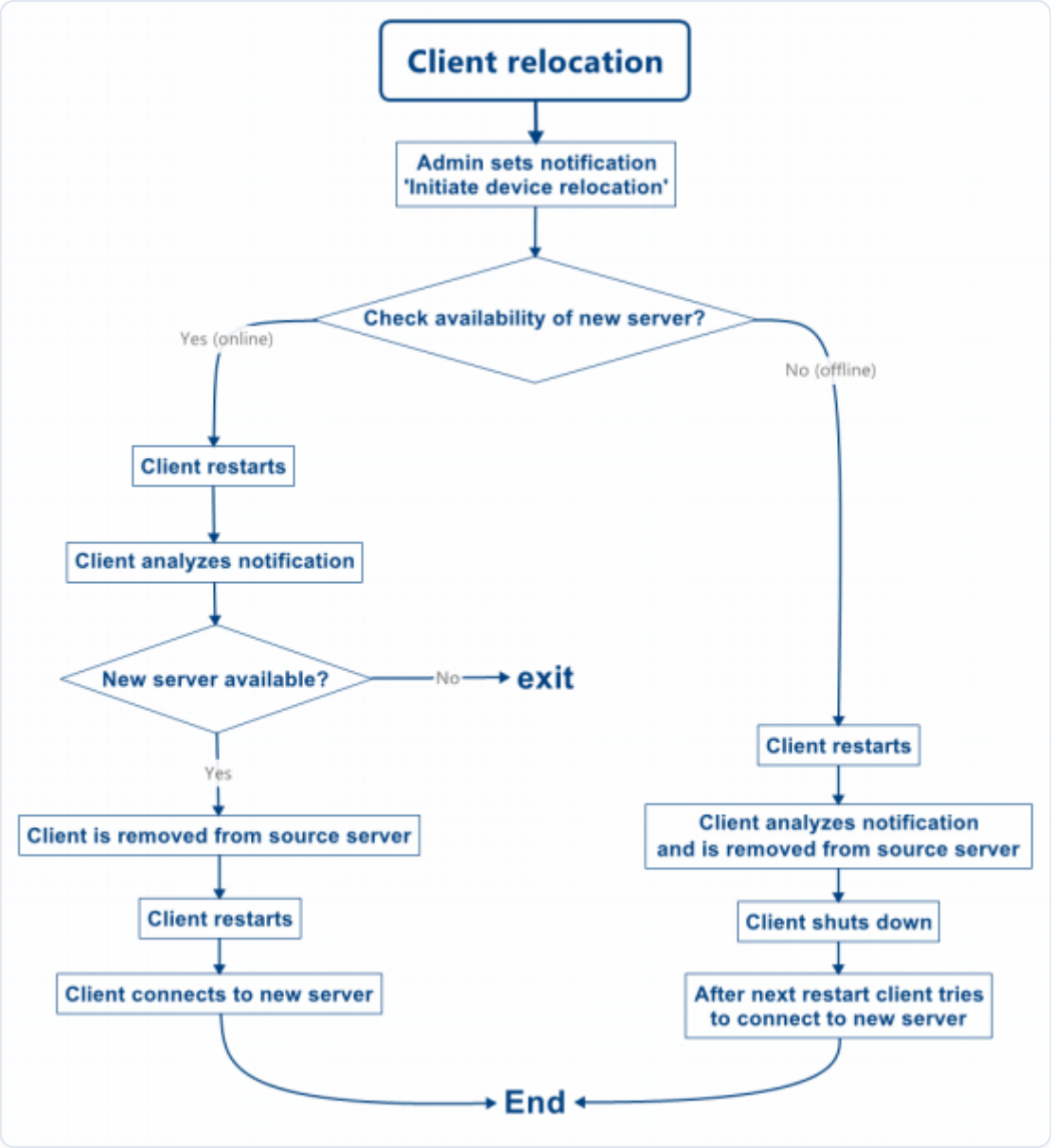
The relocation procedure is initiated by the source server (device-releasing server) and completed by the relevant devices. If you perform an offline relocation, the devices do not verify whether the target server (device-receiving server) is available and ready to accept the devices.

The administrator still triggers the relocation by setting a notification **Initiate device relocation** for the relevant devices in the Scout Console of the source server. The notification includes all required details. On the next device restart, the configuration is synchronized with the source server, the notification is analyzed and the configuration data are updated:

The relevant devices are removed from the source server without further verification. On the next restart, the devices will try to connect to the target server.

Local configuration can either be included in the transfer or left on the source server.

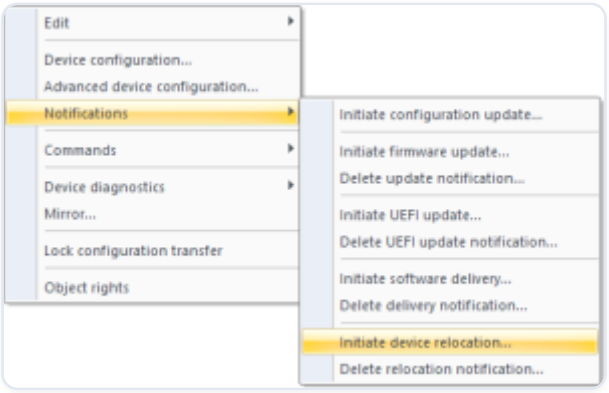
Relocation flow chart



Initiating device relocation

REQUIRES
• **Advanced Options > Devices**, clear the option **Accept only known devices**, if selected
• To ensure relocation success, the DHCP options of the device are not checked during the relocation. If, however, Scout Server DHCP options for the source server have been defined (222/223/224), on the target server, in **Device configuration > Network > LAN > Edit > Advanced select [Ignore DHCP options**

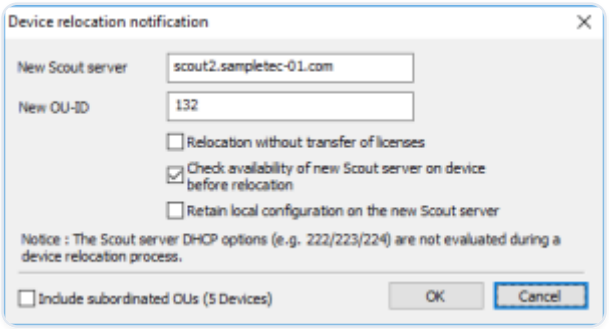
1. Select a device, an OU, a Dynamic Device Groups or devices within the **All devices** window.
2. On the context menu, click **Notifications > Initiate device relocation**.



3. In the **Device relocation notification** dialog, in **New Scout Server**, type the name (FQDN) or the IP address of the target server.

> **Note:**
>
> To connect the devices via Scout Cloud Gateway to their target server, type the name or IP address of the SCG instance. The Scout Cloud Gateway must be fully configured.

4. Edit the following fields:



| New OU-ID | ID of the destination OU on the target server | | | If you do not specify a destination OU, the devices are assigned to the default OU or to the OU defined by the OU filter rules. |

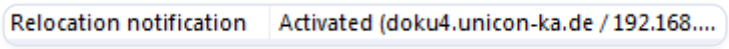
| Check availability of new Scout Server | 'online' relocation | | | Relocation is only done when the devices can access the target server via the network. | | | selected by default | | Retain local configuration on new 1[from Scout 15 2110]] | User-defined values of the local device configuration in unlocked fields will be retained. This is relevant if the corresponding option in **Advanced Options > Devices** is used. |

| Include subordinate OUs | All devices located in lower-level OUs will also be moved. |

> **Note:**
>
> Notifications are always set or deleted for all selected devices regardless of whether they are only available for individual devices.

5. Confirm the notification and confirmation.

The notifications for device relocation will be set. For the relevant devices, in the **Properties** window, the **Relocation notification** field shows the value Activated.



> **Note:**
>
> To define which fields you want to show in the **Properties** window, click .
>
> For devices not involved in the relocation, the **Relocation notification** field remains empty.

The devices evaluate the notifications after their next restart and then start the relocation process. Alternatively, trigger the relocation by command to control the time of relocation.

Scheduling the relocation process

To carry out the device relocation outside working hours, for example, and to add a slight delay after each device is processed, proceed as follows.

REQUIRES
The relocation has been initiated and the relevant devices have their relocation notification.

1. In the Scout Report Generator, identify all devices with active relocation notification.

2. Export the identified devices to a Dynamic Client Group.
3. In the Scout Console, for your Dynamic Client Group, schedule a **Restart device** command:
   - In the Command dialog, choose date and time of the execution.

- Specify a delay in milliseconds, that will be applied after the command execution of each device..

The relevant devices are restarted at defined time and then start the relocation process. They obtain their new configuration data from the target server. If you perform the relocation offline, the relevant devices will be removed from the source server, but will only connect to their new server when they can connect to it.

Deleting relocation notification

1. For the relevant devices, from the context menu, choose **Notifications > Delete relocation notification**.

2. To include the devices of all subordinate OUs, in the **Delete relocation notification** message, select the option **Include sub organization units**.

   The number of devices shown in brackets is updated dynamically.

3. Confirm with **OK**.

After refreshing the **Properties** window, you will see that the **Relocation notification** status for the relevant devices is deleted.

Notifications are always set or deleted for all selected devices regardless of whether they are only available for individual devices.

Device configuration

- Concept
- Configuration method
- Evaluating configuration data
- FollowMe Desktop
- General tab
- Network tab
- Desktop tab
- Display tab
- Mouse/Keyboard tab

- Firmware tab
- Security tab
- User authentication
- Multimedia tab
- Drives tab
- Printer tab
- Hardware tab
- Diagnostics tab
- Power management tab
- Troubleshooting device configuration

Concept

Device configuration is the key to managing a large number of devices efficiently. Configuring as many devices as possible in the same way keeps IT processes simple, and costs low. All the same, numerous different locations, heterogeneous hardware environments and additional requirements do not allow for a unified device configuration.

takes this into account by using an inheritance approach. By default, the base device configuration defined at top level passes its properties on to the devices on lower instances. The concept of inheritance helps you keep your configuration consistent and efficient. To define any variations, simply modify the relevant settings. Scout provides flexibility to override any settings on all levels.

> **Note:**
>
> Changes to the device configuration take effect on the next device restart.

> For further information, see

Inheritance of configuration

Blocking inheritance - independent device configuration

Supporting local configuration

Initiating a one-time update of all device configuration data

Accessing device configuration

Comparing device configurations between OUs or devices

Locking configuration transfer

> **Important:**
>
> The device configuration of a device depends on the software packages installed on it.

Inheritance of configuration

The base device configuration and the configuration of OUs can be inherited by lower instances.

The base device configuration is the top-level instance. Lower instances can be other OUs or individual devices.

If the option **Use parent device configuration** is active, the configuration of the superordinate element of the hierarchy is applied to the current instance. By default, the option **Use parent device configuration** is active, so a device inherits its configuration from the base device configuration.

Settings of the configuration can be edited on three levels in the Scout Console:

- Base device configuration (**Options > Base device configuration**)
- OU (**context menu > Device configuration**)
- Device (**context menu > Device configuration**)

On each level, you can inherit the device configuration from the parent level or define independent settings. To be able to override settings, you must block inheritance, that is disable the use of the parent device configuration.

> **Note:**
>
> Pay attention to the **Device configuration** dialog title. It indicates the location of the current configuration. This can be the base device configuration, or a parent OU, or the individual device.

Example: If inheritance is active and you open the configuration dialog of a device or OU subordinate to France, the title bar shows **Device configuration of OU < France > is used**. To modify any settings, open the France configuration dialog.

Supporting local configuration

User rights for modifying the local device configuration can be set for OUs and devices, even for individual fields. You can lock and disable individual fields or tabs for security reasons whereas other features such as monitor management can be allowed. For further information, see User rights.

If individual (local) configuration is allowed, make sure that the relevant configuration data are prevented from being overridden when the Scout configuration is reloaded on the next device restart.

RETAINING LOCAL DEVICE CONFIGURATION

1. Click **Options > Advanced options > Devices**.
2. Under **Field update**, select **Retain local configuration (unlocked fields)**.

When the Scout device configuration data are reloaded, only locked tabs and fields are updated. Local user configuration data in unlocked fields are kept.

RETAINING LOCAL DEVICE CONFIGURATION DURING FACTORY RESET

1. Select the option **Advanced options > Retain local configuration (unlocked fields)**, see above.
2. In the command dialog for the factory reset, select **Retain local configuration (unlocked fields)**.

The device is reset to its initial state and the device configuration of the locked fields is reset. However, the local user configurations in unlocked fields are retained.

For further information, see Factory reset command.

Initiating a one-time update of all device configuration data

For example, in case of a defective user configuration, however, the administrator, in the Scout Console, can set a flag to override all device configuration data on the next device restart.

1. In the Scout Console, for the relevant device or OU, from the context menu, choose **Notifications > Initiate configuration update**.
2. For OUs, in the notification message, select whether you want to apply this on all subordinated devices (default).[1[from Scout 15 2209]]

The relevant devices will be marked to obtain a copy of the relevant Scout device configuration data including unlocked fields, on the next device restart.

Comparing device configurations between OUs or devices

The device configuration of different OUs or devices can be compared by using a dedicated window.

1. Click **View > Window > Compare configuration**.

   The window Compare configuration is shown as a permanent window in the lower part of the console window.

2. Drag two or more OUs or devices into the **Compare configuration** window by using a drag-and-drop operation.

   Or:

   On the context menu of the relevant OU or device, click **Edit > Add to configuration compare**.

3. On the icon bar of the **Compare configuration** window, click the icon.

   The device configuration of the listed OUs or devices is compared. Differences in the main properties are shown.

4. To view all of the information, on the icon bar of the **Compare configuration** window, click the icon.

All properties are shown.

> **Note:**

> To compare actual and target settings of individual devices, use a report. For further information, see Evaluating configuration data.

Locking configuration transfer

Individual devices can be excluded from the process of updating device configuration data.

REQUIRES
Object right Activate/Lock configuration transfer

1. For the relevant device, open the context menu.
2. Click the option **Lock configuration transfer**.

The device is no longer provided with updated device configuration data but remains in the Scout management.

> **Note:**
>
> For newly added devices without management license, the configuration transfer is automatically locked.

Configuration method

During system start of the devices managed by the Scout Enterprise Management Suite®, the devices connect to their Scout Server and check whether any updated configuration data are available, taking inheritance into account. There may be updates for the following data:

- Device configuration
- Application definition
- Files configured for transfer
- Advanced file entries

The Scout Server identifies the relevant configuration data at run-time. That means that all modifications made to configurations in the Scout Console until then are included.

- Configuration run

Configuration run

If the system determines updated configuration data for a device, the relevant modified data are identified, compressed and saved to the database. They are then transferred to the device in one step.

When modifying the configuration of a large number of devices (e.g. when changing the application definitions due to a move to another back-end infrastructure), the administrator can initiate the process of identifying and compressing the relevant data in a data object in advance, for instance at night. To do so, the administrator uses the **Configuration run** command. With this command, the required configuration data can be prepared to be ready for transfer on the next device restart, which might be on the next working day.

PERFORMING A CONFIGURATION RUN

> **Note:**
>
> The configuration run command only prepares configuration data for devices with a configuration delta.

1. For the relevant OU or Dynamic Device Group, from the context menu, choose **Commands**.

2. On the sub-menu, click the **Configuration run** command.
3. Specify a time for execution and confirm with **Schedule**.

*The processing progress is shown in the **Command history**.*

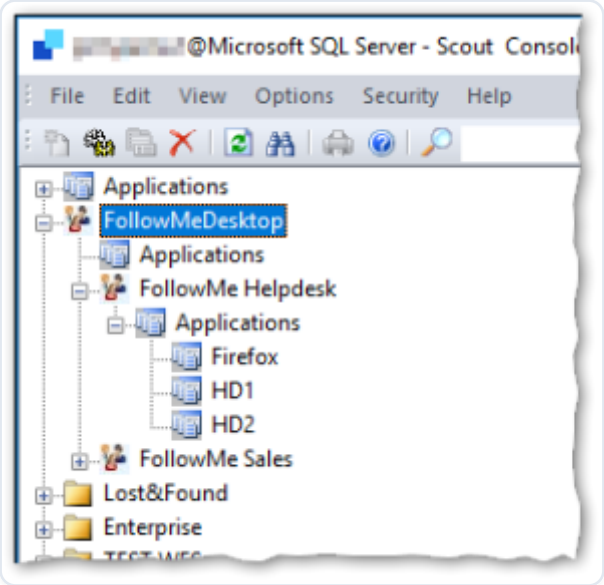
FollowMe Desktop

> **Note:**
>
> To use this feature, you will need user authentication via Active Directory.

> FollowMe Desktop allows you to make user-specific configuration settings that are valid across devices.

In addition to the device configuration option, which is bound to devices, the FollowMe Desktop feature allows users to take their configuration settings with them to any device. Users' configured desktop layout and configured applications "follow" them to any device they log on to.

FollowMe Desktop is configured in the tree structure node of the same name and is implemented as one of the top-level OUs. This instance can be used right away or can act as a container for further FollowMe configurations, which are added as subordinate OUs. This is how a device-independent and cross-hierarchical structure of configurations is mapped, which might be motivated by users' subject-specific functions. For example, a **FollowMe Helpdesk** OU could contain all application definitions and the desktop layout for Helpdesk employees.



A FollowMe Desktop configuration may consist of application definitions and defined configuration values of the device configuration. For further information, see Scope of configurable options.

The FollowMe configurations are assigned to users via AD properties. The administrator applies filter definitions to the FollowMe Desktop instance and all subordinate OUs, which then filter for the relevant AD membership.

When an AD user logs on to a device that is enabled for FollowMe Desktop, the user's AD properties are matched against the filter definitions of the FollowMe OUs. If a FollowMe configuration with matching filter values is found, the desktop is loaded with the layout and applications defined in that FollowMe OU. When working with the FollowMe configuration, users can make changes to it, such as placing application icons freely on the desktop. These changes, however, will not be saved.

After the user closes his session and logs off, the original device configuration is reloaded.

FollowMe Desktop configurations can be exported and imported.

- Configuring FollowMe Desktop
- Scope of configurable options

Configuring FollowMe Desktop

To use the FollowMe Desktop feature, make one or more FollowMe configurations available to certain AD users via filter definitions. Then define on which devices you want to allow the retrieval of a FollowMe configuration.

CREATING FOLLOWME CONFIGURATIONS

REQUIRES
Administrator base right **Show FollowMe Desktop OU**

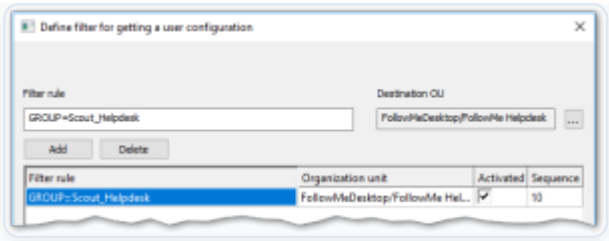
1. In the tree structure, below the **FollowMe Desktop** top-level OU, create further FollowMe OUs such as a **FollowMe Helpdesk** OU. To do so, from the context menu, choose **Add > Organization unit**.
2. Edit the device configuration of the new FollowMe OUs (Desktop, Drives, USB options, Power management). Then add the relevant application definitions to the FollowMe OUs.

> **Note:**
>
> New FollowMe Desktop OUs do not inherit configuration values from the base device configuration or base applications. Within the FollowMe Desktop structure, however, inheritance is active by default. Subordinate FollowMe OUs inherit the device configuration and applications from the parent FollowMe instance, unless you disable inheritance.

> *The configuration values of a FollowMe-OU overwrite the device configuration values valid on the device.*

3. For each FollowMe OU, define a filter that filters for an AD membership. To do so, from the context menu, choose **Set filter**.

> **Note:**
>
> The filter definitions are independent of any inheritance. For each FollowMe OU, define a separate filter.



In the filter rule, you can filter for the AD properties **User**, **Group** or **Domain**. Syntax: USER|GROUP|DOMAIN=<Value\> Multiple filter rules are allowed.

> **Note:**
>
> For devices with the FollowMe Desktop enabled, the user AD properties are written to the eluxd.log file (search for FollowMeDesktop).
>
> *All active filter rules are processed in the specified order. AD users matching this filter definition will have access to the applications and desktop properties configured in the FollowMe OU after logging on to devices for which FollowMe Desktop is enabled.*

User-related FollowMe configurations are defined. In the next step, define for which OUs or devices these configurations may be retrieved.

ENABLING DEVICES FOR FOLLOWME DESKTOP

Specify the devices that shall have access to multiple configurations. In some cases, users may prefer the device configuration defined in the OU structure on their workstation device. On other devices, for example in the service area, they may prefer a FollowMe configuration.

In the device configuration of an OU / device (or in the base device configuration), you define whether a device should retrieve Follow-Me configurations when authorized AD users log on.

- In the device configuration of the relevant devices, under **General > FollowMe Desktop**, select **Request configuration for logged-on users**.

After AD users log on, their AD properties are evaluated. If the Scout Server finds a matching FollowMe OU, the configuration defined there is loaded.

Scope of configurable options

The following properties can be configured for FollowMe OUs:

- All application types with all application properties
- Software default settings for Firefox and Chromium (browser home directory)
- Device configuration with following options

	OPTIONS
Desktop	All options except date and time

	OPTIONS
Desktop > Advanced	Advanced Sort Configuration panel
	System bar
	Background picture
Drives	All options
Hardware	All USB options
Power management	All options

Configuration values of a FollowMe OU will overwrite the configuration values of the "normal" device configuration on this device. For options not listed, the values of the "normal" device configuration remain active.

Note:

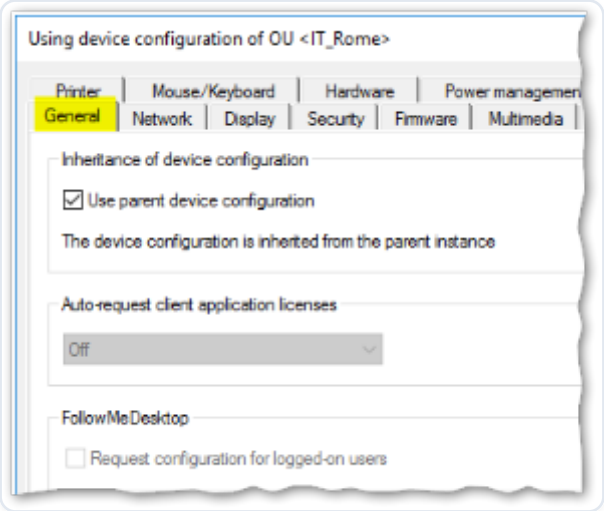
Within a FollowMe OU, only those applications will be available that have been defined for it.

The administrator can further limit the scope of the configurable options of FollowMe OUs for operational administrators (**Security > Manage administrators > Default object rights**).

General tab

Use parent

This option of the **General** tab is active by default and ensures that consistent device configurations are used. If **Use parent device configuration** is selected, all other fields of the dialog are disabled: For the relevant device or OU, the device configuration is used that is displayed in the title bar (in the figure IT_Rome). This is where an administrator will normally make any configuration changes.



Note:

In some cases, it might be useful to disable the **Use parent device configuration** option temporarily.

For further information, see Blocking inheritance - independent device configuration.

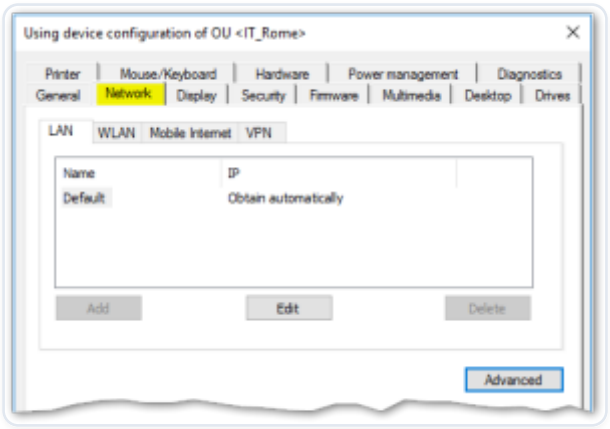
Additional options

OPTION	DESCRIPTION
[FollowMe Desktop] / Request configuration	Allows authorized users to use multiple configurations on this device. For further information, see Configuring FollowMe Desktop.

Hardware information for individual devices is listed in the **Properties** window of the Scout Console. For further information, see Properties.

Network tab

Depending on the installed image and the hardware used, users can use different network connections. In the Scout Console, you define network profiles for different network types. The user can then choose from the defined network connections on the system bar.



The following network profiles are available:

- LAN (only one profile, cannot be deleted)
- Wireless LAN
- Mobile Internet (Wireless Wide Area Network)
- VPNfrom Scout Enterprise Management Suite® 15 2101 For earlier versions, see VPN in the Advanced device configuration.

To create VPN network profiles, use the Advanced device configuration for individual devices. For further information, see VPN.

- Defining a LAN profile
- Advanced network settings
- Defining a WLAN profile
- WPA support
- Corporate WLAN
- Defining a Mobile Internet profile (WWAN)
- Defining a VPN profile
- Options for all network profiles
- Proxy configuration
- Internet Protocol version 6 (IPv6)
- Firewall for eLux devices

Corporate WLAN

A corporate WLAN providing access to internal resources can be secured by 802.1X with firewall policies tailored to specific needs.

After having set up your WPA configuration file, you can deploy it wherever required. For further information, see WPA support.

With a corporate WLAN, you can allow users to create their own WLAN profiles in parallel. For example, a mobile device could use the provided LAN connection when it is attached to the docking station on the job, but change automatically to the corporate WLAN when undocked. Once the device is started in the home office, eLux® connects to the manually configured WLAN.

For further information, see Adding a WLAN profile in the **eLux** guide.

Internet Protocol version 6 (IPv6)

In addition to full support of Internet Protocol Version 4 (**IPv4**), **IPv6** is used by default for local applications including automatic network configuration (DHCP, DNS, NTP).

- To disable IPv6, in the Scout Console, for the relevant devices, configure the following Advanced file entry:

File	/setup/terminal.ini
Section	Network
Entry	DisableIPv6
Value	true (default = false)

For further information, see Advanced file entries.

Firewall for eLux® devices

To secure your eLux devices with a firewall, for example to allow exactly one more connection besides the connection to the Scout Server, use the eLux **Firewall support** package, which allows you to define appropriate rules.

By default, **nftables** are used.Alternatively, the **iptables** syntax can be used.

Once the eLux **Firewall support** package is installed, the firewall rules are applied on system start.By default, only data packets that are required for communication with the Scout Server are allowed to pass.To allow further connections, define filtering rules and transfer them to the devices in a file.

If the **Firewall support** package has been installed on the device without filtering rules, the firewall will only start if the feature package **Strict firewall policy** is installed.Then communication

between the Scout Server and the device is established via the management protocol (port 22125) and no other communication is allowed.

CONFIGURING FIREWALL RULES (NFTABLES)

> **Note:**

> The eLux package **Firewall support** and the included feature packages **eLux firewall plugin** and **firewall nftables programs and libraries** must be installed on the devices. This may require modifications of the image definition file on the web server via ELIAS.

1. Create the nftables.conf file according to the following example:

   ```
   table ip filter {chain input {tcp dport 22 accepttcp sport 80 accept}
   chain output {tcp sport 22 accepttcp dport 80 accept}}
   ```

   In the example, outgoing http and incoming ssh connections are accepted.

2. Transfer the files to the devices to /setup/firewall/nftables.conf.To do so, use the Scout feature **Files configured for transfer**. For further information, see https://docs.citrix.com/en-us/unicon-elux-scout/2611/admin_guides/introduction/advanced_options/advanced_files.htmlin the **Scout** guide.

CONFIGURING FIREWALL RULES (IPTABLES)

> **Note:**

> The eLux package and the included feature packages and must be installed on the .This may require modifications of the image definition file on the web server via ELIAS.

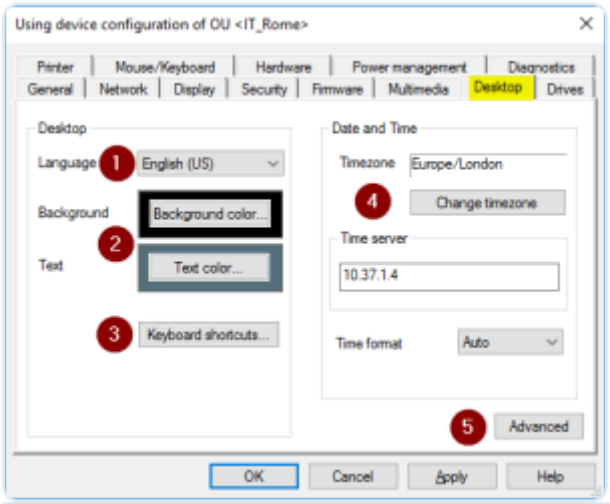
1. Create the file rules.v4 for IPV4 or rules.v6 for IPV6.Use the **iptables** syntax for the rules.
2. Transfer the files to the devices to /setup/firewall/rules.v4 or /setup/firewall/rules.v6, respectively.To do so, use the Scout feature .

   For further information, see https://docs.citrix.com/en-us/unicon-elux-scout/2611/admin_guides/introduction/advanced_options/advanced_files.htmlin the **Scout** guide.

All rule files are included in the **Diagnostics** feature and are part of the System template.

Desktop tab

On the **Desktop** tab, you can modify the eLux® desktop layout.



1	Display language
2	Background and text colors
3	Keyboard shortcuts
4	Date and time settings
5	Advanced settings

- Configuring language and colors
- Keyboard shortcuts
- Date and time
- Advanced desktop settings
- eLux RP 6 User Interface

Configuring language and colors

1. For the relevant device or OU, open **Device configuration > Desktop**.
2. In the **Language** list, click the preferred desktop and application language.

   The following languages are supported: English, German, French and Spanish

> **Note:**

> The language setting refers to the display of desktop elements. It does not affect text services and input. For a smooth performance, ensure that the applications support the selected language.

3. Click the **Background color** button to select a desktop background color.
4. Click the **Text color** button to select a text color for the application icons. Make sure there is sufficient contrast to the background color.

For further configuration options of the eLux RP 6 desktop, see eLux RP 6 User Interface.

Keyboard shortcuts

For switching applications and locking the screen, keyboard shortcuts are already predefined and you are free to customize them. To define keyboard shortcuts freely, follow the syntax rules given.

DEFINING KEYBOARD SHORTCUTS

1. Open **Device configuration> Desktop > Keyboard shortcuts**.
2. You may select keyboard shortcuts for the following actions:

OPTION	DESCRIPTION	HOW TO DEFINE
Switch applications	Switch between open applications or sessions	Select an option from the list-field.
	The default shortcut Alt+Ctrl↑ helps avoid conflicts with the shortcut Alt+Tab which is used to switch between the tasks within a Windows session.	
Log off	Log off currently logged-on user (AD users)	Specify the desired key combination as free-text, see below.
	The logon dialog is then displayed.	
Lock screen	Activate password-protected screen saver (AD users)	Specify the desired key combination as free-text, see below. Otherwise, the

OPTION	DESCRIPTION	HOW TO DEFINE
		default key combination is active.
	Default: Ctrl+Alt+End	

Furthermore, you can define a key combination for multi-monitor environments that allows users to quickly switch between clone mode and extended desktop.[1[from Scout 15 2101 and eLux RP 6 2101]] For further information, see Multiple monitors.

RULES FOR SPECIFYING KEY COMBINATIONS

- Key combinations consist of a combination of one or more modifier keys and a single non-modifier key
- For the non-modifier key, you can choose from the following keys: Letter keys, number keys, function keys, Windows logo keys, key, position and numpad keys as specified
- Key combinations must be specified in the following format:

```
<modifier key\><modifier key (optional)><modifier key (optional)>additional key
```

No spaces or other characters may be placed between the keys.

- The spelling of the key names must follow the specification, see examples. To receive a complete list of allowed key names and their spelling, use the following command in an eLux® shell:

Modifier key names: Non-modifier key names:

Examples

KEY COMBINATION	DESCRIPTION
Ctrl+Alt+Escape	
Shift+Ctrl+l	
Mod4+Alt+F1	Mod4 corresponds to the Windows logo key
Ctrl+Mod4+Alt+End	

KEY COMBINATION	DESCRIPTION
Mod4+Super_R	(= right Windows button if used as key) Super_R corresponds to the Windows logo key on the right if used as a key
Mod5+KP_End	Mod5 corresponds to Alt Gr, KP_End corresponds to the End key of the numpad

Important:

If you define a key combination for eLux that is already defined within an application/session, this key combination will only work for eLux. Avoid using the same key combinations in different environments.

BEHAVIOR OF THE CAPS LOCK KEY

Note:

May be configured via user interface from , see Advanced mouse and keyboard settings.

In most environments, pressing the () key in combination with letter keys results in the display of uppercase letters, while the number keys above the letter block output numbers despite the key. To display the special characters of the number keys, the key must be pressed.

- To let users write special characters instead of numbers while pressing , for the relevant devices, configure the following Advanced file entry:[2[from eLux RP 6 2107]]

File	`/setup/terminal.ini`
Section	`Keyboard`
Entry	`ForceShiftLock`
Value	true, Default: `false`

For further information, see Advanced file entries.

The Caps Lock key then behaves like the Shift key.

Date and time

OPTION	DESCRIPTION
Time zone	Click [Change time zone] and select the required time zone from the list.
Time server	Under [Time server], specify the relevant server name or IP address.
	The time server must comply with the Network Time Protocol (RFC 1305) or the Simple Network Time Protocol, a simplified form of NTP. Microsoft Windows operating systems include the [W32Time] service which communicates via SNTP in older versions such as Windows 2000, and uses NTP in later versions. The time service is started automatically.
	The service runs on port 123 and uses the UDP protocol.
	For further information on the Windows Time Service, see the Microsoft documentation.
	For further information on NTP, see http://www.ntp.org.
Time format [from Scout 15 2101 and eLux® RP 6 2209]	The time can be displayed in 24 hour or 12 hour time format.
	• Auto (default): The displayed time format depends on the configured display language (see same dialog).
	• 12 hour

OPTION	DESCRIPTION
	• 24 hour

Advanced desktop settings

SYSTEM BAR

OPTION	DESCRIPTION
Show system bar	The system bar will be displayed on the devices. Below, select its behavior and the icons you want to show on the system bar. (selected by default)
Hide automatically	The system bar is hidden by default. As soon as users point the mouse to the bottom of the screen, the system bar is displayed.
Show Desktop icon	Users click this icon to minimize all open windows and show the desktop. (selected by default)
Show live information icons	These icons allow users to view current status information such as plugged USB devices via right-click. (selected by default)
Show time	Shows the current time. When users point the mouse to the time, the current date is shown. (selected by default)
Show Config panel icon	Allows users to open the device configuration (Configuration panel). (selected by default)

Important:

Only when the Configuration panel is displayed, can the administrator unlock the configuration via device password locally on the device.

QUICK CONFIG: QUICK ACCESS TO CONFIG PANEL DIALOGS VIA SYSTEM BAR

All options are selected by default.

OPTION	DESCRIPTION
Volume	Volume control for input and output devices
Keyboard	Keyboard language and key speed
Display	Screen settings
Peripherals	Settings for peripherals such as USB and Bluetooth devices, and COM ports
Network	Network information and setup, disconnect/ connect
Device information	Information on the device
Date and time	Date and time settings

BACKGROUND

Different background images can be defined for primary/secondary monitors and for the time before/after the AD logon.

- Select a background image from the list-field. Then configure it via the buttons.

OPTION	DESCRIPTION
Background picture	Desktop wallpaper (default). If further pictures are defined, this one will be used only for the primary monitor and after AD logon.
Additional picture	Desktop wallpaper for secondary monitors
Background picture AD	Desktop wallpaper until AD logon (primary monitor)
Additional picture AD	Desktop wallpaper until AD logon (second and more monitors)

OPTION	DESCRIPTION
Load	Browse the file system and select a picture file. The picture file will be imported into the database.
	• The following file formats are supported: [.svg], [.png], [.jpg] (Note: .jpg cannot be previewed)
	• Maximum file size: 500 KB
Delete	Remove the current background image from the database.
Set URL	As an alternative to a picture file from the file system, specify a web address for loading pictures.

Note:

Make sure you have enough flash memory on the devices. The background image is stored in the directory of the flash card.

FURTHER OPTIONS

OPTION	DESCRIPTION
Sort Configuration panel	The Configuration panel dialogs are displayed in alphabetical order (active by default)
Timer for shutdown confirmation	Before the device is shut down, a message is displayed for the specified time period (in seconds). This allows users to prevent the shutdown by clicking the **Cancel** button or pressing the ESC key. (deselected by default) This option has its own user right and object

OPTION	DESCRIPTION
	right (for administrators). Both rights are enabled by default.
Desktop sort order	Sort order of the desktop icons. The administrator can specify the sort order for users without the user right **Sorting desktop icons**. Users who have this user right can freely place desktop icons on the desktop.

eLux® RP 6 User Interface

eLux RP 6 devices come with a new desktop interface that can be customized to your needs and that provides a personal desktop view. For further information, see eLux RP 6 User Interface in the **eLux** guide.

CUSTOMIZING THE LAYOUT OF THE ELUX RP 6 USER INTERFACE

1. For the relevant OU, use the Advanced file entries feature of the Scout Console to modify the client file in the section. Add the following new entries and specify the relevant values:

ENTRY	VALUE RANGE	DESCRIPTION
DesktopLogo	none	Option 1 replaces the eLux Logo in the lower right section by the specified picture file. Example: Option 2 removes the eLux Logo in the lower right section.

> **Note:**
>
> To show an individual picture file, transfer the picture to the devices. For further information, see Files configured for transfer.

ENTRY	VALUE RANGE	DESCRIPTION
DesktopTextColor	#<>	Text color of application icons

ENTRY	VALUE RANGE	DESCRIPTION
DesktopHighlightedTextColor	#<>	Text color of application icons when the mouse pointer is moved over them
DesktopTitleTextColor	#<>	Color of the folder/tab title (All Applications, StoreFront store name)
DesktopSearchTextColor	#<>	Text color in the search field
DesktopSearchBackgroundColor	#<>	Background color of the search field
DesktopSearchIconColor	#<>	Color of search icon (magnifier)
DesktopSortIconColor	#<>	Color of sort icon (A-Z)

2. To display a background image, configure the relevant picture file in the **Advanced** desktop settings of the device configuration. For further information, see Advanced desktop settings.

   For multiple monitors, the image will be used as a background image on each monitor.

> **Note:**
>
> The background image overrides the **Background** color defined on the **Desktop** tab.

3. To set a solid background color for the desktop, use the **Background** option on the **Desktop** tab.

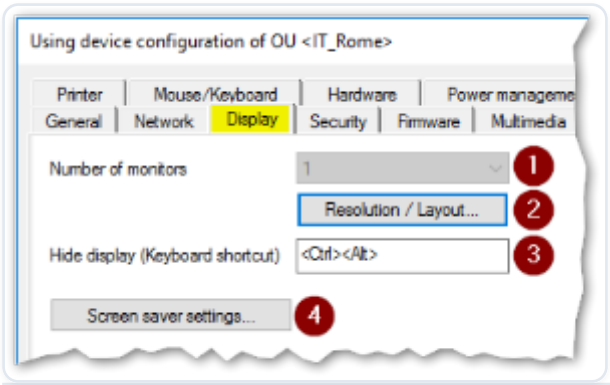
   To set a gradient background color, use the RGB values .

4. To configure the position of the system bar, define the following entry in the file in the Layout section:

ENTRY	VALUE RANGE	DESCRIPTION
SystembarPosition	0/1	0:bottom of the screen
		1: top of the screen

Display tab

On the tab, you can choose between display settings and screen saver settings. To define display settings for one or more monitors, open the dialog.



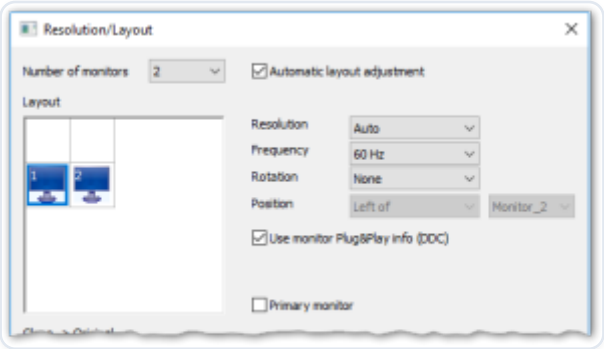
1	Define the number of monitors together with further display options in the [Resolution/ Layout] dialog.
2	Configure the display
	• Display options per monitor
	• Multiple monitors
3	Keyboard shortcut to hide/show the content of a monitor
4	Select and configure a screen saver

Note:

Power saving options can be found on a dedicated tab, see Power management tab.

Configuring the display

1. If you have defined more than one monitor, in the **Resolution/Layout** dialog, select a blue monitor icon.



2. For the selected monitor, use the list fields on the right to specify the screen resolution, frequency, and rotation.

Resolution	Screen resolutions that are not listed may be added to the database table `dbo.Resolution`. After modifying the table, restart the Scout Console.
Frequency	Refresh rate
Rotation	The screen display can be rotated 270° (left), 180° (inverted) and 90° (right).
Position	Only for multiple monitors

3. To have the values supported by the monitor processed by the device, select **Use monitor Plug&Play Info (DDC)**.

   Clear the option to activate the **Monitor class** field.

> **Note:**
>
> If you decide to use adapters or the analog VGA port to connect monitors to devices, warranty for the operation of these devices will be excluded. These types of combinations are not part of functional acceptance tests.

4. To define the selected monitor as the primary one, select **Primary monitor**.

5. Confirm with **OK** and **Apply**.

> **Important:**
>
> If your monitors do not support the settings you have defined, you may have to perform a factory reset of the device and modify the desired screen settings.

Multiple monitors

Up to eight monitors can be configured.

DEFINING MULTIPLE MONITORS

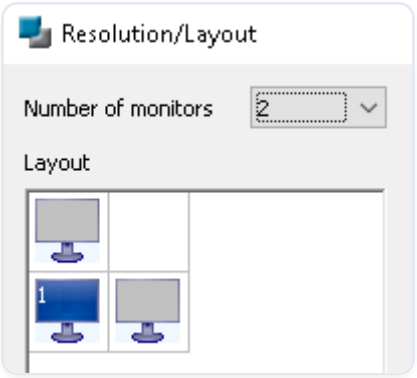
1. In **Device configuration > Display**, click **Resolution/Layout** to open the dialog of the same name.

   The field **Number of monitors** specifies one monitor by default. In the field below, this monitor is represented by a blue monitor icon with the number 1. By default, the first monitor is defined as the primary monitor (see option in the lower section).

   For a different setting, see the instructions below.

2. In the **Number of monitors** list, select how many monitors you want to connect to the devices.

   Once you have defined more than one monitor, their possible positions (horizontal and vertical) are shown as gray monitor icons.

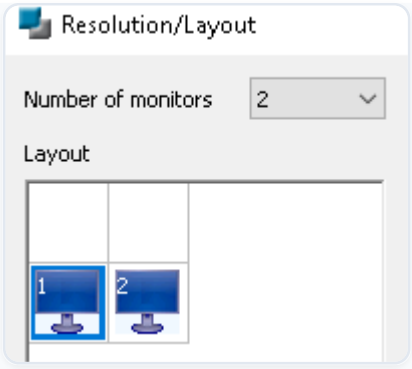


3. Double-click the gray monitor icon that shows the position of the second monitor.

   The selected monitor icon is shown in blue with the number .

> **Note:**

Alternatively, right-click a monitor position to assign a monitor to it.



4. If you have specified more than two monitors, double-click the desired gray monitor icons, one after the other.

   Each of the defined monitors is shown as a blue monitor icon with its number.

5. To automatically adjust the layout after one of the monitors is removed, select the option **Automatic layout adjustment**.

   If the option is not active, the current layout is retained regardless of the actual situation.

Note:

A four-monitor configuration is supported on the following devices: Dell Z50QQ, Hewlett-Packard t620 Plus and Hewlett-Packard t730. A five-monitor configuration is supported on the following devices: Fujitsu FUTRO S940 and Fujitsu FUTRO S9010.

DEFINING POSITIONS OF ALL MONITORS FREELY

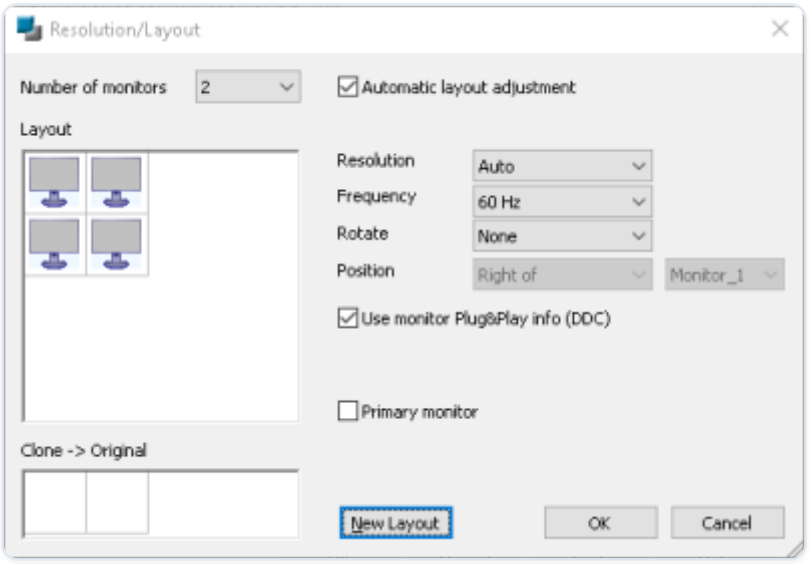
To define the position of the first monitor, use a new layout.

1. In the **Resolution/Layout** dialog, in the **Number of monitors** list, select how many monitors you want to connect to the devices.

   The first monitor is shown as a blue monitor icon. For each additional monitor, their possible positions (horizontal and vertical) are shown as gray monitor icons.

2. Click **New layout**.

   For the number of monitors selected, all possible positions are shown as gray monitor icons:

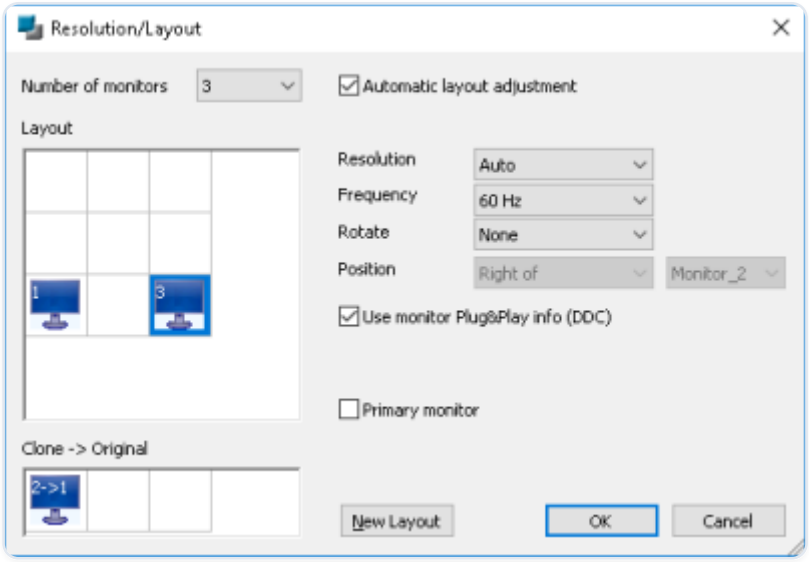


3. Double-click the desired monitor position for the first monitor. Then, double-click the desired monitor positions of the remaining monitors.

CLONING MONITORS (CLONE MODE)

If you have specified more than one monitor, by default the system configures the monitors as extended desktops (one continuous desktop over all monitors). Alternatively, for individual monitors, after having defined them, you can activate the Clone mode (same display on multiple monitors):

- Right-click the last defined blue monitor icon, and from the context menu, choose **Clone of** .



To deactivate the Clone mode, use the **New Layout** feature (see above).

KEY COMBINATION FOR SWITCHING BETWEEN CLONE MODE AND EXTENDED DESKTOP

-- from Scout und --

- To allow users to quickly switch between the two modes, define the following Advanced file entry for the relevant devices:

File	`/setup/terminal.ini`
Section	Hotkeys
Entry	CloneDisplays
Value	`<Mod4>p`
Value	`<Mod4>` corresponds to the Windows logo key

For further information, see Advanced file entries in the **Scout** guide.

Screen saver

CONFIGURING THE SCREEN SAVER

1. Under **Display > Screen saver settings**, choose between a black screen, a specific screen saver or multiple screen savers.
2. Depending on the option chosen, select one or more screen savers from the list. To select multiple entries, press `Shift` or `Ctrl`

> **Note:**
>
> The **HTML** option allows you to choose a website.

3. To configure each screen saver, use the settings on the right.

ENABLING THE SCREEN SAVER

- On the **Power management** tab, for each profile, select **Enable screen saver after** and specify a waiting time in minutes.

LOCKING THE SCREEN ON THE DEVICE

If the screen saver is enabled, eLux® users can lock the screen before the configured waiting time with the following key combination:

- Press CTRL+ALT+END
- Password-protected screen saver
- Downloading picture files for screen saver

PASSWORD-PROTECTED SCREEN SAVER

Note:

When user authentication is enabled, password protection of the screen saver becomes active and cannot be turned off.

The password is set to $ELUXPASSWORD. For further information, see Where to apply user variables.

The screen saver becomes active after the defined time period and the system is locked. By pressing a key or moving the mouse, a dialog is displayed for unlocking. It provides the following options to users:

OPTION	BUTTON	DESCRIPTION
The logged-on user unlocks the screen by entering his/her password / smart card	Unlock	Default
Another person leaves a message for the logged-on user	Message	The screen remains locked. The logged-on user receives a notification with the message when he or she unlocks the screen. This function is enabled by default and can be disabled by the following Advanced file entry: File: [/setup/terminal.ini], section: [xscreensaver_dialog], entry: [MessageEnabled], value: [false]
Another user authenticates to log off the previous user (and to log on), restart or shut down the device.	Log off	Useful if devices are used by multiple users: Allows users to reuse devices that have been left without logging off and therefore are blocked. Once the new user has authenticated, the **Restart**,

OPTION	BUTTON	DESCRIPTION
		Shut down and **Log off** buttons become active. In any case, the previously logged-on user is logged off. This function is disabled by default and can be enabled by the following Advanced file entry: File: [/setup/terminal.ini], section: [xscreensaver_dialog], entry: [ShowSysCommandButtons], value: [true]

Important:

Data loss may occur if the **Log off** option is used followed by a restart, shutdown, or logoff. The user currently logged on is logged off regardless of whether the documents or data last edited have been saved.

DOWNLOADING PICTURE FILES FOR SCREEN SAVER

Optionally, Scout administrators may configure direct download of picture files to the devices into the screen saver's picture directory. To do so, use the **FileFetch** tool, which downloads graphic files via **wget**.

REQUIRES
• The eLux® package **FileFetch** must be installed on the devices. This may require modifications of the image definition file on the web server via ELIAS.
• The picture files must be located in the specified directory and have the file extension .gif, .jpg, or .png. The filename must be numeric. Examples: 0001.jpg, 0002.jpg, 0003.png, 0004.gif

- For the relevant devices, configure the web server and the directory of the picture files.To do so, choose the **Advanced file entries** feature of the Scout Console:

File	[/setup/terminal.ini]
Section	[FileFetch]
Entry	[URL]
Value	URL of the web server including path
Example	[http://webserver.sampletec-01.com/eluxng/pictures]

For further information, see Files configured for transfer.

The **FileFetch** tool checks on each device restart whether new picture files are available on the web server for download.

Hide display

-- from Scout --

Users can temporarily hide the content of a monitor using a predefined key combination. This can be helpful when multiple monitors are used and certain information should not be visible to others present.

The following options are provided:

OPTION	DEFAULT KEYBOARD SHORTCUT
Hide display of primary monitor	Ctrl > Alt > 1
Hide display of second and all other monitors	Ctrl > Alt > 2

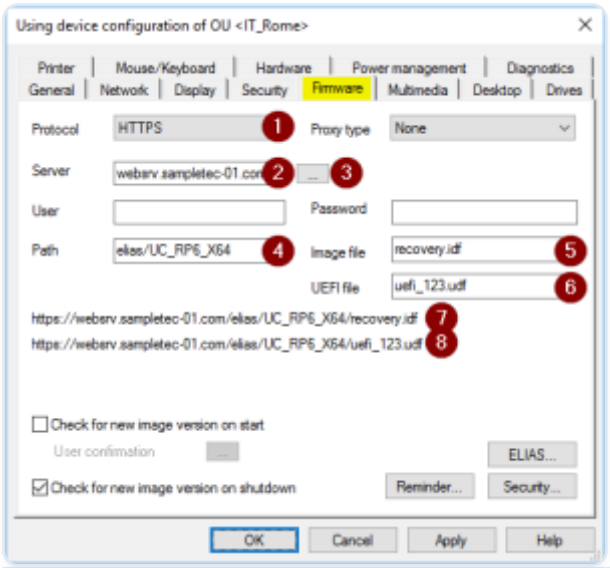
The digits **1** and **2** are fixed. The modifier keys Ctrl > Alt may be defined differently by you.

DEFINING ALTERNATIVE MODIFIER KEYS

- Next to **Hide display (Keyboard shortcut)**, enter the desired modifier keys. For further information on how to spell the key names, see Keyboard shortcuts.

Firmware tab

The **Firmware** tab provides the relevant information required to perform a firmware update (software update) of the devices via network.



1	Network protocol for software package transfer from the web server to the devices
2	Name or IP address of the web server providing the eLux® software packages and the image files
3	Optional: alternative web server for devices connected via VPN[\^1]
4	Directory path of eLux software packages on the web server (container path)
5	Image file on the web server, also called image definition file or IDF, defines the software packages to be installed on the devices
	The image is created with the ELIAS application and then made available on the web server.
6	Optional: UEFI file in eLux container with assignment of device types and UEFI firmware to be installed[\^2]
7	From 1-4 a URL is generated, which is used by the devices to update the firmware.

	If the web or FTP server is password-protected, the user name and parametrized password are also included in the URL.
8	If a UEFI file is specified, the system generates a URL which is used by the devices to update the UEFI system.

Note:

The image file and container path can be parametrized if required.

The image file, UEFI file and container path can be predefined globally by the administrator so that you may select them from the respective list-field. For further information, see Predefined IDFs and containers.

- Configuring firmware updates
- Protecting firmware configuration
- Different hardware models
- Different eLux versions
- Update to new partition layout (1.77 / 1.84 GB to 2.35 / 2.41 GB)
- Update to new partition layout (2.35 / 2.41 GB to 3.75 GB)
- Firmware security through signature
- Update deferment by users

Protecting firmware configuration

Image definition files (IDF) are provided on the web server in an eLux® container. They must be specified in the device configuration under **Firmware** in order for the devices to access the intended image in the event of an update request. Depending on the object rights defined, in the **Firmware** configuration, administrators are allowed to enter individual IDF names as free text or need to select one of the predefined IDFs from the list-field. The same applies to the software container (**Path** field) in the **Firmware** configuration.

To protect such critical firmware configuration parameters, the IDFs and the container paths to be selected or configured for a firmware update can be defined in advance. In combination with the relevant object rights, operational administrators can then only choose between predefined values.

For and later versions, the firmware of UEFI systems[1[from eLux RP 6 2107]] can be updated via the same mechanism as the software (firmware update). Therefore, a **UEFI file** field can be found in the same dialog that behaves accordingly.

SETTING OBJECT RIGHTS FOR FIRMWARE CONFIGURATION FIELDS

The object rights for the **Image file**, **Path**, and **UEFI file** fields are each divided into **predefined** and **user-defined**. If you grant an administrator both rights, he can add new entries as free text as an alternative to selecting a predefined entry from the list-field.

REQUIRES
Administrator policies are enabled.

1. For the relevant OU, from the context menu, choose **Object rights...**
2. Select the relevant administrator /administrator group and click **Edit object rights...**
3. For **Device configuration > Firmware**, change the object rights as required by double-clicking or pressing the `Space` bar:

   | Image file (predefined) | The administrator can only select one of the IDFs provided in the **Image file** list-field on the **Firmware** tab. | | | The list-field contains predefined IDFs (see below). If predefined IDFs are missing, the list-field shows the recently used IDFs. |

   | Image file (user-defined) | The administrator is allowed to enter any IDF name into the text field. |

   | Path (predefined) | The administrator can only select one of the paths provided in the **Path** list-field. | | | The list-field contains predefined paths (see below). If predefined paths are missing, the list-field shows the recently used paths. |

   | Path (user-defined) | The administrator is allowed to enter any path into the text field. | | | The path must correspond to a software container on the web server. |

   | UEFI file (predefined) | The administrator can only select one of the files provided in the **UEFI file** list-field. | | | The list-field contains predefined UEFI files (see below). If predefined files are missing, the list-field shows the recently used UEFI files. |

| UEFI file (user-defined) | The administrator is allowed to enter any UEFI file name into the text field. |

4. Confirm with **OK**.

For further information, see Administrator policy.

PREDEFINING FIRMWARE CONFIGURATION VALUES

1. On the menu, click **Options > Advanced options > Predefined IDFs**.
2. To add additional IDF names, click the **Add** button and edit the new entry. Note that the spelling must match the actual names.
3. For all entries you want to share in the firmware configuration, select the **Valid** option.
4. Confirm with **Apply** and **OK**.

All valid IDFs, container paths and UEFI files are provided in the device configuration under **Firmware** and can be used by authorized administrators.

> **Note:**
>
> Scout does not check the physical existence of files or container paths on the web server.
>
> For further information, see Predefined IDFs and containers.

Different hardware models

-- from eLux® RP 6 2103 --

To update the firmware of devices with an alternative image depending on the hardware model, the release parameter is available. For example, you can update newer model types to a CR version, while older models remain on their previous (LTSR) image. If there are no changes to the previous image, as in the LTSR image example, all devices whose type is not whitelisted are unaffected by the update command.

For the release parameter, in the firmware configuration, an `__RM__` string is inserted into the image file name.

Before an update is executed, an appropriately configured device resolves the parameter using a whitelist:

- If the model type of the device is part of the whitelist, the `__RM__` parameter is replaced by a string you define. So the device will pull the alternative image with the newly created name.
- If the model type of the device is **not** part of the whitelist or the whitelist cannot be loaded, the `__RM__` parameter is removed from the image name. This leaves the device on the previous image (image name as in the firmware configuration, but without the `__RM__` string).

CREATING A WHITELIST

REQUIRES
The web server must support the file extension [.mee] in the MIME type settings.

1. Create a text file named `elux.mee`. Then, enter the section name `__RM__`.

   > **Note:**
   >
   > Make sure you use the correct spelling for the section name: Two underscores followed by `__RM__` (uppercase) followed by two more underscores.

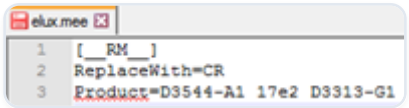
2. Begin the second line with `ReplaceWith=` and then define a short string.

   This string must be included in the image name for the alternative image, see below. If you do not specify anything, the string `CR` is set by default.

3. Begin the third line with the string `Product=`. Then, enter all model types you want to receive the alternative image in the same line.

   Separate the model types by white spaces.

   Enter type names that contain white spaces without the white spaces.

   You can retrieve the model type of a device from the Scout Console. It is shown in the **Properties** window in **Asset > Hardware information > Type**. On the devices, the model type can be found in the `terminal.ini` under `HWInfo.Product`.

4. Copy the `elux.mee` file into your `UC_RP6_X64` container on the web server.

PREPARING THE SOFTWARE CONTAINER ON THE WEB SERVER (ELIAS)

1. Leave the existing image as you want the devices outside the whitelist to receive it.

Example: `recovery.idf`

2. In ELIAS, create an alternative image that you want the devices in the whitelist to receive.

   For example, to be able to update newer models to a CR version, create an image containing the eLux packages of the latest CR version.

3. Assign the same name to the alternate image, but include the string defined in the whitelist under `ReplaceWith=`

   Example: `recoveryCR.idf`

The software container now contains two images whose names differ only by the defined string, and the whitelist `elux.mee`

MODIFYING DEVICE CONFIGURATION

1. For your OU, open the device configuration under **Firmware**.
2. Under **Image file** insert the `__RM__` string into the file name, see screenshot above. The file extension `.idf` must be kept.

   Example: `revovery__RM__.idf`

   > **Note:**
   >
   > Make sure you use the correct spelling: Two underscores followed by (uppercase) followed by two more underscores.

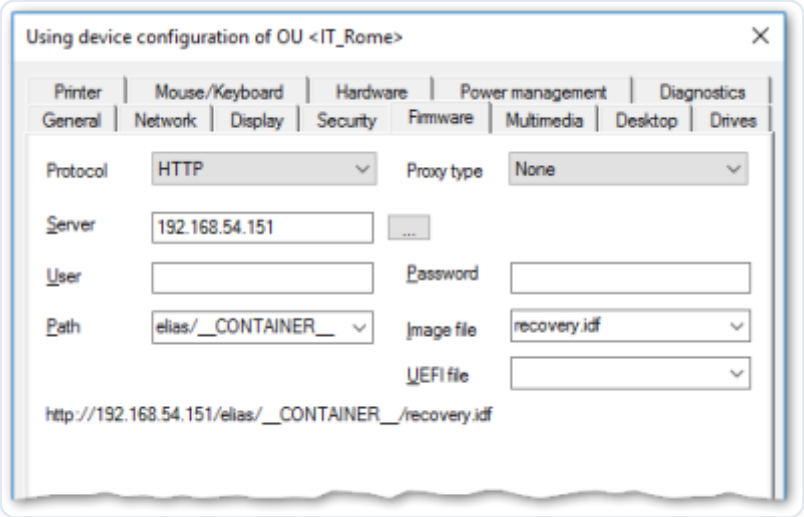
3. Confirm with **Apply** and **OK**.

The next update command on your OU will apply the alternative image to the whitelisted devices.

Different eLux® versions

The `__CONTAINER__` parameter is useful if you have more than one major eLux version in use and you want to update the firmware of multiple devices regardless of their major version. All updated devices remain within their respective major version.

The parameter replaces the container path in the URL



The `__CONTAINER__` parameter is part of the directory path and parametrizes the relevant software container (directory) on your web or FTP server. When the administrator sends an update command to a device configured with a container parameter, the parameter is resolved by a macro according to the installed eLux version.

Example:

If you run devices with both, eLux RP 6 and eLux 7, the eLux RP 6 devices require the `UC_RP6_X64` container, while the eLux 7 devices require the `UC_ELUX7` container. To provide all devices with their appropriate software, in Device configuration > Firmware > Path of all devices , use the `__CONTAINER__` parameter. The devices then resolve the container parameter according to their installed major version to `UC_RP6_X64` or `UC_ELUX7`, respectively. The advantage is that an image file with the same name can be used for both platforms. So, in ELIAS, you would create an image with the same name for eLux RP 6 and for eLux 7.

> **Note:**
>
> In some cases, it can be useful to replace the container parameter by a fixed container name. In this case, the entry in the Path field must correspond to the container name on the web server.

ELIAS 18 CONTAINERS

To use the container parameter with ELIAS 18, choose the following names for your containers:

MAJOR ELUX VERSION	CONTAINER NAME
eLux RP 6	UC_RP6_X64

MAJOR ELUX VERSION	CONTAINER NAME
eLux 7	UC_ELUX7

For further information, see Creating a container in the **ELIAS 18** guide.

SPELLING OF THE CONTAINER PARAMETER

When replacing a fixed container name by the container parameter, make sure you use the correct spelling:

Two underscores followed by the word `CONTAINER` (all uppercase) followed by two more underscores.

> **Note:**
>
> You can use the container parameter in the firmware configuration and in the recovery settings **Options > Recovery setttings**.

Update to new partition layout (1.77 / 1.84 GB to 2.35 / 2.41 GB)

Earlier eLux RP 6 versions up to eLux RP 6 2104 LTSR have created smaller system partitions during installation than current eLux versions. Beginning with eLux RP 6 2107, the system partition is created with 2.35 GB with encryption and with 2.41 GB without encryption, so that larger images can be included.For further information, see eLux RP 6 partitions.

For updating the firmware of devices with eLux® RP 6 2104 LTSR to the current eLux RP 6 version,[1[from eLux RP 6 2110]] the system partition must be repartitioned before the actual update installation. In order to still be able to perform an update installation in one step, the solution described below uses two images to which devices are updated, one after the other. The second firmware update is automatically triggered.

> **Note:**
>
> When you start from eLux RP 6 2104 LTSR CU3 or CU4, the intermediate and final image will be based on the latest eLux version and you can follow the instructions below. When you start from eLux RP 6 2104 LTSR CU1 or CU2, the intermediate image must be based on a different eLux version. In this case, please contact our support for more details.

TWO IMAGES AS PREREQUISITES

	IMAGE WITH PARTITION RESIZE PARAMETER	FINAL IMAGE
Size	Corresponds to the old world with a maximum of 1.77 GB	May have a size over 2 GB
Content	• Current **BaseOS** package	• All packages from the first image
	• Specific eLux software package named **Update Support Utility** which automatically triggers the second update on the devices	• Additional packages that did not fit into the first image (optional)
		• No **Update Support Utility**-package
Name	Must contain the _PR__ parameter as a string so that the macro can be executed later on	Same name as first image, but without the _PR__ string
	Example: [recovery_PR__.idf]	Example: [recovery.idf]

Both images must be in the same container.

PROCEDURE

For the update installation of your devices you only start the update to the first image with **Partition Resize** parameter. After that it's the system's turn:

- The first update process installs the current eLux version without running the macro. The devices now have the first image on board. After installation, the **Update Support Utility** package triggers another update.

- On the second update request, the current eLux version on the devices identifies the **Partition Resize** parameter and runs the macro:
- The *_PR* string will be removed from the update URL.
- An update to the final image will be performed.

> **Note:**
>
> The automatic update to the final image is always performed without formatting the system partition. This is independent of whether the option was set during the first update.

- The **Update Support Utility** package will be uninstalled because it is not included in the final image.

CONFIGURING FIRMWARE UPDATES WITH PARTITION RESIZE PARAMETER

1. In ELIAS, create an image with **Partition Resize** parameter. Follow the specifications from the table above. The image file name must contain the string *PR on any position. Example: recovery__PR_*.idf
2. In ELIAS, in the same container, create a second image which will be used as the final image. Follow the specifications from the table above. The image name must match the first image name, but must contain the string *_PR*. Example: recovery.idf
3. In the Scout Console, for the relevant OU, open **Device configuration > Firmware**. In the **Image file** field, enter the name of the first image (the one with *_PR* parameter).

   The image file specified in the figure above requires the existence of a second image named which is the final image.

   Edit the other fields of the **Firmware** tab. For further information, see Configuring firmware updates.
4. For the relevant OU, perform a firmware update.

The devices update to the first image with current eLux version. Then, the **Update Support Utility** package triggers a second automatic update that installs the final image on the devices. This automatic update is retried on each system restart until a successful installation uninstalls the **Update Support Utility** package.

For devices connected via VPN, the automatic update is triggered only after the VPN tunnel is established. The installation is performed without user interaction.

SPELLING OF THE PARTITION RESIZE MACRO STRING

Make sure to use the correct spelling:

Two underscores followed by the string PR (all uppercase) followed by two more underscores.

> **Note:**

> You can use the **Partition Resize** parameter in the firmware configuration or in the recovery settings **Options > Recovery setttings**.

Update to new partition layout (2.35 / 2.41 GB to 3.75 GB)

eLux® RP 6 2302 LTSR CU2 contains a modified partition layout that provides more space on the system partition for additional software packages. To upgrade to eLux RP 6 2302 LTSR CU2, please note the following information.

SYSTEM REQUIREMENTS FOR THE NEW PARTITION LAYOUT

- Memory size at least 8 GB
- System partition size 2.35 GB / 2.41 GB

Devices with less than 8 GB memory size can be updated to eLux RP 6 2302 LTSR CU2, but the repartitioning will not be performed.

DOWNGRADE TO THE PREVIOUS PARTITION LAYOUT

Once the repartitioning is complete, it is no longer possible to downgrade to an eLux version with the previous partition layout via firmware update. Please use PXE recovery or USB stick recovery if a downgrade is required.

PERFORM THE FIRMWARE UPDATE

The update to eLux RP 6 2302 LTSR CU2 is done exclusively via a specially configured firmware update, which automatically triggers a second update. The image macro *_P4* and the service package **Custom updater** (part of the eLux RP 6 2302 LTSR CU2 release) are used for this purpose.

The process is started by a firmware update to an image that has the image macro `__P4_` in its name. This `_P4__` image must contain the package and must not be larger than 2.35 GB / 2.41 GB.

After the installation of the *_P4* image on the device a restart is performed and the **Custom updater** starts the second firmware update automatically.

The second update installs the final image, which must have the same name, but without the image macro *_P4* in the name and without the package **Custom updater** in the image. The size of this final image can be up to 3.75 GB.

PROCESS

a. Currently installed image with system partition 2.35 GB / 2.41 GB Example: MyCurrentImage.idf

b. Firmware update to 2 LTSR CU2 image with image macro *P4 in its name and the **Custom updater** package Example: MyNewImage__P4_*.idf / max. 2.35 GB / 2.41 GB

c. Automatic update to the final eLux RP 6 2302 LTSR CU2 image without image macro and without **Custom updater** package Example: MyNewImage.idf / max. 3.75 GB

> **Note:**

> The image with the image macro *P4_* in the name can be kept in the firmware tab of the device configuration until the update to the next eLux image. eLux RP 6 2302 LTSR CU2 ignores the macro and always checks the update against the final image without image macro.

INSTALLING THE UPDATE STEP-BY-STEP

1. Create an IDF for eLux RP 6 2302 LTSR CU2 with the image macro *P4_* in the name containing the package **Custom updater**. Image size <= 2.35 GB / 2.41 GB. Example: MyNewImage__P4__.idf
2. Via ELIAS perform **Save as** for this IDF and remove the image macro *P4_* from the name and remove the package **Custom updater** from the image. Add more packages if needed. Image size <= 3.75 GB. Example: MyNewImage.idf
3. In **device configuration > Firmware** tab enter the name of the image with the image macro *P4_* in its name. Example: MyNewImage__P4__.idf
4. In Scout Console perform a firmware update via **update command** or **update notification**.

   Recommendations:

- Send the **update command** only to devices with the status "Switched on" (see parameters of the update command) or send a separate command to switch on the units before the update command. Otherwise, the devices could be switched off after the first update.
- Using **Check for new image version on start / on shutdown** may result in a failed second update on some devices and is therefore not recommended.

CHECK THE RESULTS

In Scout Console > **Properties** after a successful update to the final image UEFI devices should display the following values:

- Partition info | system 3,75 GB | setup 256,00 MB | update 3,75 GB (up to 20,00 GB)
- Image | Image name without image macro `__P4__` in its name Example: MyNewImage.idf

> **Note:**
>
> Due to a missing boot partition, legacy BIOS devices should display the Partition info | system 4,00 GB | setup 255,00 MB | update 3,75 GB (up to 20,00 GB).

TROUBLESHOOTING

ERROR / PROBLEM	REASON	SOLUTION
Properties in Scout Console	The second automatic update via **Custom updater** was not carried out	Restart the device
• Partition info / system 4,07 GB or 4,17 GB / setup 120,00 MB /		
• Image / `MyNewImage_P4__.idf`		
• Image name with image macro `__P4__` in its name		

ERROR / PROBLEM	REASON	SOLUTION
Properties in Scout Console	A second update was carried out, but not via the **Custom updater**.	Perform a manual update via Scout Console with a format of the system partition before update.
• Partition info / system 4,07 GB or 4,17 GB / setup 120,00 MB		
• Image / `MyNewImage.idf`		
• Image name without image macro `__P4__` in its name		

Firmware security through signatures

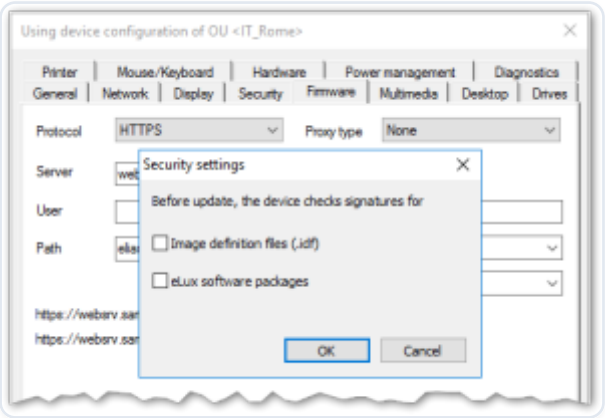
You can configure the firmware configuration in the or on the device to have the device check signatures each time before an update is performed. An update is then only performed if the signature of the image definition file (IDF) and/or the signature of the eLux software packages have been successfully verified. The update cannot be run, however, if the IDF or one of the eLux software packages to be installed does not have a valid or verifiable signature.

> **Important:**

> A signature check of eLux software packages requires an update partition on the device. On devices without an update partition, signatures can only be checked for image definition files but not for eLux software packages. For further information on update partitions, see eLux partitions.

ACTIVATING SIGNATURE CHECK

1. In the , under **Device configuration > Firmware**, click **Security** On the eLux® RP 6 device, select **Configuration panel > Firmware > Check signatures before update**.



2. Under **Signature check before update**, select the **Image definition file** option and/or the **eLux software packages** option.
3. Confirm with **OK** and **Apply**.

> **Note:**

> In eLux, both options are provided in the Config panel, under **Firmware**.

> The signature verification results are documented in the update log file on the device. After an update has been performed, the update log file is sent to the Scout Server. To view it for the selected device, in the **Properties** window, double-click the **Update status** field.

CERTIFICATES

Verifying the IDF signature on the client side requires the root certificate, but also the signature certificate in the local device directory /setup/cacerts. If you use own certificates for signing IDFs or individually composed eLux packages, configure their transfer to the devices. To do so, use the Scout feature **Files configured for transfer**. For eLux packages provided by Unicon, all required certificates are included in the BaseOS.

> **Note:**

> When updated code signing certificates are made available on our technical portal, download and import them into ELIAS. Instructions are included.

> For further information on how to create image signatures, see Signing an image in the ELIAS 18 guide.

Security tab

1	Define end-user rights to functions
2	Configure Identity provider and logon procedure for users
3	Configure mirror sessions
4	Configure kiosk mode

Identity Providers

As of Scout version 15 2507, the following Identity Providers can be configured for eLux® 7

IDENTITY PROVIDER	INFORMATION
None	No authentication required and the eLux desktop will show after boot
Active Directory	Provide AD related information
Active Directory + Smart Card	Provide AD and Smart Card related information
Citrix®	Provide Citrix related information
Entra ID	Entra ID authentication for the native AVD client. From UniconOS 7.2611, this sign-in and the AVD client use a new Microsoft Entra application, see Moving the AVD client to the new Entra application.
Evidian	Provide Evidian related information
Imprivata	Provide Imprivata related information

Apart from these authentication methods, which can be configured in the UI, you can also configure `Entra ID` and `OKTA` authentication for eLux 7 using the advanced file entries.

Configuring Entra ID authentication for eLux 7

FILE	SECTION	KEY	VALUE
`/setup/terminal.ini`	`Security`	`AuthType`	`12`
`/setup/terminal.ini`	`OpenId`	`AuthorizeEndpoint`	`<URL>`
`/setup/terminal.ini`	`OpenId`	`RedirectUrl`	`<URL>`
`/setup/terminal.ini`	`OpenId`	`ClientId`	`<ID>`

Configuring OKTA authentication for eLux 7

FILE	SECTION	KEY	VALUE
`/setup/terminal.ini`	`Security`	`AuthType`	`12`
`/setup/terminal.ini`	`OpenId`	`AuthorizeEndpoint`	`<URL>`
`/setup/terminal.ini`	`OpenId`	`RedirectUrl`	`<URL>`
`/setup/terminal.ini`	`OpenId`	`ClientId`	`<ID>`
`/setup/terminal.ini`	`OpenId`	`ClientSecret`	`<Secret>`

Configuring mirroring

1. On the **Security** tab, under **Mirroring**, select **Enable**.

> **Note:**

> After you enable mirroring, the device needs two restarts to be able to start the VNC server.

2. Click **Settings** for configuration:

OPTION	DESCRIPTION
Password	If you define a mirror password, the password will be requested before a mirror session can be started. (optional) The password must have 6 characters minimum and 8 characters maximum.
Read-only access	Allows read access only. If not selected, in the mirroring session, the user may still select the option so that the administrator has read-only access.
User must confirm within	Before the mirror session can be started, the user must confirm. Specify how many seconds you want to display the request before the connection is aborted.
Log mirror session	For each mirror session, a log file is created and saved to a sub-directory of the files directory.
Encrypt data transmission	Uses encrypted transmission.
Allow from only	Mirroring is allowed only from Scout. The device shows and logs which administrator requested the session, and it refuses VNC viewers that connect to it directly.
Log off on disconnection	Automatic logoff as soon as the connection is aborted.

3. Confirm with **OK** and **Apply**.

For further information, see Mirroring.

> **Note:**

The user can cancel a mirror session at any time.

User authentication

User logon and authentication can take place directly on the eLux® device after device start-up or on the back-end server to which the device connects.

eLux supports user authentication with Active Directory. This can be done via username and password or via smart card. Both eLux methods are configured on the tab of the device configuration as described below. Additionally, authentication via Evidian is supported.

Note that for authentication under eLux, the appropriate eLux software packages must be installed on the devices. For smart card check, in addition to the eLux functions, appropriate middleware and hardware drivers for the smart card readers are required.

Additional options for AD users

If Active Directory is used for user authentication, users log on with their AD domain account and password on their device. Pass-through logon is supported by applications which provide access to back-end systems (Citrix®, RDP, VMware).

On the device, in the Configuration panel under **Information**, the user logged on is shown.

CHANGE PASSWORD

To change the password, users choose the eLux® command **Logoff** and then click **Change password**:

PASSWORD-PROTECTED SCREEN SAVER

Note:

When user authentication is enabled, password protection of the screen saver becomes active and cannot be turned off.

The password is set to $ELUXPASSWORD. For further information, see Where to apply user variables.

The screen saver becomes active after the defined time period and the system is locked. By pressing a key or moving the mouse, a dialog is displayed for unlocking. It provides the following options to users:

OPTION	BUTTON	DESCRIPTION
The logged-on user unlocks the screen by entering his/her password / smart card	Unlock	Default
Another person leaves a message for the logged-on user	Message	The screen remains locked. The logged-on user receives a notification with the message when he or she unlocks the screen. This function is enabled by default and can be disabled by the following Advanced file entry: File: /setup/terminal.ini, section: xscreensaver_dialog, entry: MessageEnabled, value: false
Another user authenticates to log off the previous user (and to log on), restart or shut down the device.	Log off	Useful if devices are used by multiple users: Allows users to reuse devices that have been left without logging off and therefore are blocked. Once the new user has authenticated, the **Restart, Shut down and Log off** buttons become active. In any case, the previously logged-on user is logged off. This function is disabled by default and can be enabled by the following Advanced file entry: File: /setup/terminal.ini, section: xscreensaver_dialog, entry: ShowSysCommandButtons, value: true

Important:

Data loss may occur if the **Log off** option is used followed by a restart, shutdown, or logoff. The user currently logged on is logged off regardless of whether the documents or data last edited have been saved.

SERVICE APP

You can enable AD users to start eLux in service mode. To do so, define a service app that can be started from the logon dialog using the **Service** button. For further information, see Defining a service app.

STARTING CONFIGURATION PANEL FROM LOGON DIALOG

You can enable AD users to access the Configuration panel before they log on. This allows them to connect to a WLAN or VPN or to change the language before logging on.

You can show the system bar for users authenticating via Evidian before they log on. This allows them to access the Configuration panel and Command panel.

- For the relevant devices, under **Security > User rights** on the bottom, enable the user right **Start Config panel from logon dialog**.

AD: On the next restart, the logon dialog shows the button for opening the Configuration panel , so AD users can access the options you allow them to.

Evidian: While the Evidian authentication manager is displayed, users may open the Configuration Panel dialogs and access commands such as **Shut down** and **Restart** in the Command Panel.

Note:

By default, the user right **Start Config panel from logon dialog** is disabled.

User variables

Note:

> If you want to use user variables, the **User authentication modules** and **Open LDAP** packages must be installed on the devices. This may require modifications of the image definition file on the web server via ELIAS.

> The values of user variables are used by the authentication server for the log-on process. User variables can also be used in some fields of the eLux control panel.

Predefined user variables are

$ELUXUSER

$ELUXDOMAIN

$ELUXPASSWORD

The variables are used when users log on and user authentication is active.

Where to apply user variables

Defining new user variables

WHERE TO APPLY USER VARIABLES

> **Note:**

> To use this feature, user authentication via Active Directory is required.

> When they are applied, user variables must have a leading $. User variables can be applied in the following fields:

Device configuration

	FIELD	USER VARIABLE
Drives	Username	\$ELUXUSER
	Password	\$ELUXPASSWORD
	Directory, Server, Share	Any \$ELUX variable
	Browser home directory	Any \$ELUX variable

	FIELD	USER VARIABLE
Power management	Enable screen saver	\$ELUXPASSWORD
	(also manual activation)	\$ELUXPASSWORD

Application definition

CATEGORY	FIELD	USER VARIABLE
Citrix®	Server	Any $ELUX® variable
RDP	Username	$ELUXUSER
VMware Horizon	Password	$ELUXPASSWORD
	Domain	$ELUXDOMAIN
Browser	Proxy type, Proxy port	Any $ELUX variable
Tarantella	Server	Any $ELUX variable
Local / Custom application	Parameter for all programs run from the command line	Any $ELUX variable
	Example: eluxrdp / vint.sampletec-01.com.de /u:\$ELUXUSER /p:\$ELUXPASSWORD	

DEFINING NEW USER VARIABLES

> **Note:**
>
> To use this feature, user authentication via Active Directory is required.

> You can define your own user variables as local variables based on LDAP attributes. The variable definition has the form Local variable = LDAP variable

1. On the **Security** tab, under **User authentication**, select **Active Directory (AD)** or **Active Directory + Smartcard**.

2. Click **Edit**.
3. Under **User authentication > User variables**, edit the following fields:

OPTION	DESCRIPTION
Local variable	The name of the local variable must begin with the string `ELUX` (but without `$`), which can be followed by any characters. Example: `ELUXFULLNAME` , More than one entry can be transferred if you append a `#` sign to the variable name. Example: `ELUXmemberOf#`
LDAP variable	To be able to use the LDAP variables, the relevant LDAP variable names are assigned to the individual variable as an attribute. Example 1: `ELUXFULLNAME = displayName` , Example 2: `ELUXmemberOf# = memberOf` , If there are several `memberOf` values within the search base on the authentication server, they are assigned to the local variables `ELUXmemberOf_1`, `ELUXmemberOf_2` and so on.

4. Confirm with **OK** and **Apply**.

> **Note:**
>
> User variables are defined without a leading $, but when they are applied they must begin with $.

multimedia tab

The audio **output** devices are grouped in classes depending on their connector:

USB	USB port
Analog	TRS audio jack (phone connector) or integrated devices

Digital	DisplayPort or HDMI

For each device class, you can control the volume level and **Mute** option separately.

By default, the priority is defined: USB -- Analog -- Digital.

- To change priority, move the list entries by using drag-and-drop operations.

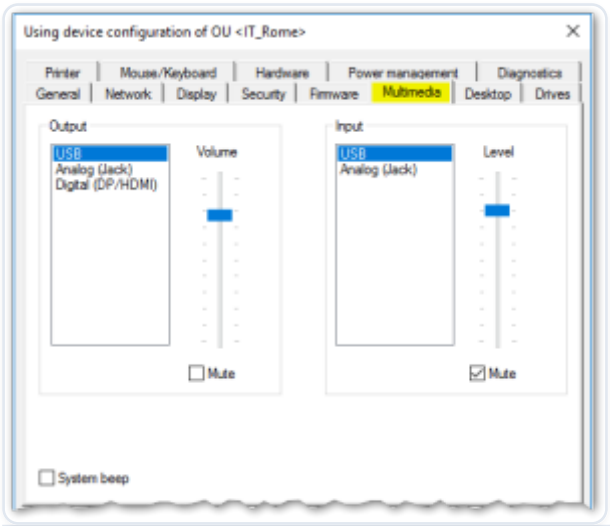
The audio **input** devices are grouped in classes depending on their connector:

USB	USB port
Analog	TRS audio jack (phone connector) or integrated devices

For each device class, you can control the sensitivity and **Mute** option separately.

By default, the priority is defined: USB -- Analog.

- To change priority, move the list entries by using drag-and-drop operations.



OPTION	DESCRIPTION
Volume (Output)	Slider to control the playback sound level for the selected device class (0 to 100)
Sensitivity (Input)	Slider to control the level of sensitivity for recording for the selected device class (0 to 100)

OPTION	DESCRIPTION
Mute (Output and Input)	No sound is reproduced / recorded
System beep	Acoustic feedback signal when switching off the device
Priority of device classes	By using drag-and-drop operations, you can change the priority of the device classes for input and output. The top entry has the highest priority.

Drives tab

Define shared network directories on your Windows server as drives that can be accessed by the devices. Any drive defined this way can for example be used as browser home directory.

- Defining a network drive
- Mount points

Defining a network drive

1. In **Device configuration > Drives > SMB Drives**, click **New**.
2. Edit the following fields:

OPTION	DESCRIPTION
Directory	Any name for the directory
Server	Name of the server including the path
Share	Windows share name
Username and password	Windows username and password to access the directory
Domain	Can alternatively be specified in the **User** field: `<DomainUser>` or `<User@Domain>`
AD authentication (only Scout)	The Active Directory logon data are used to access the directory. The fields **Username** and **Password** are disabled.

OPTION	DESCRIPTION
Test (only eLux®)	Checks if the network share can be accessed with the specified data

Note:

To access network drives with AD authentication, the software package **Network drive share** and the included feature package **Linux Key Management Utilities** must be installed on the devices. This may require modifications of the image definition file on the web server via ELIAS.

3. Click **OK** and **Apply**.

The directory path /smb/ is automatically inserted before the directory name.The data are provided on the local flash drive under `/smb/<Directory name>`.

Example: /smb/share

Note:

Here, you may apply LDAP user variables.For further information, see Where to apply user variables.

To make browser settings such as bookmarks permanently available, define a network drive as the browser home directory.For further information, see Browser home directory.

Mount points

Mount points are used to access local resources through an application. The following mount points are provided by eLux®:

Samba	/smb
NFS	/nfs

Internal CD-ROM	/media/cdrom
USB devices	/media/usbdisk

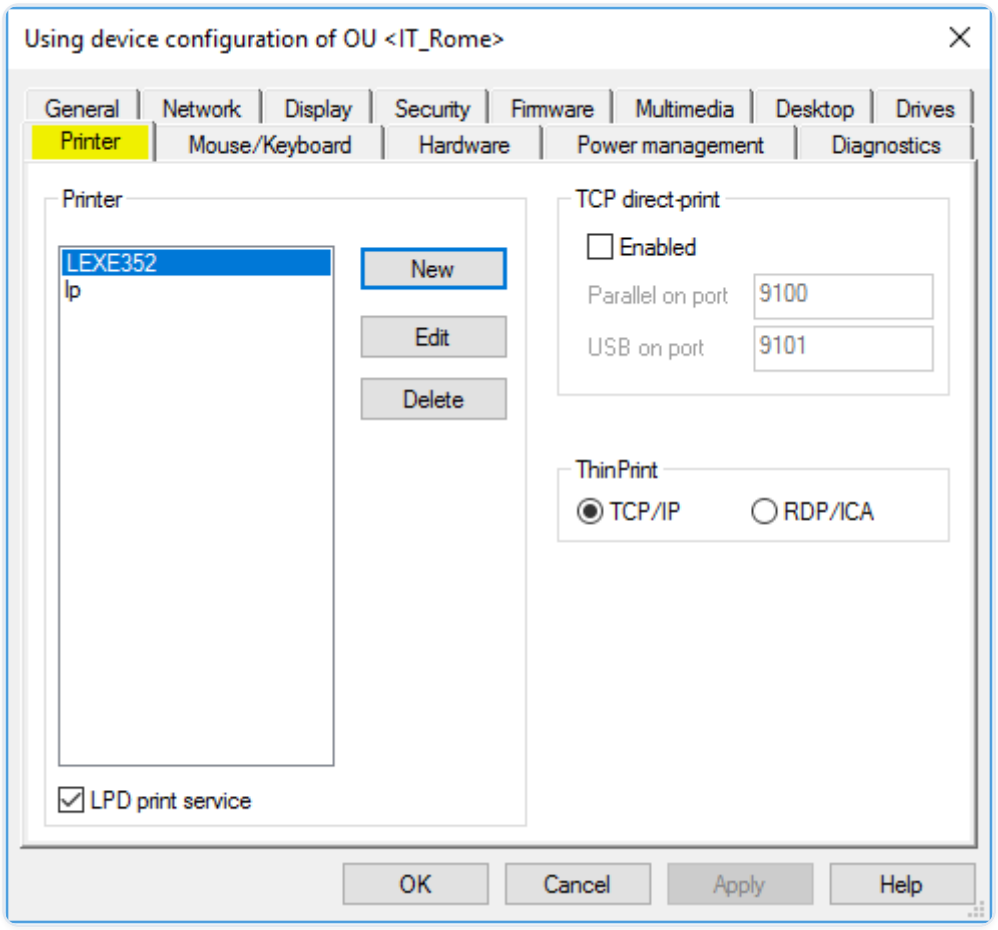
For USB devices, mount points are assigned chronologically: The first device is assigned /media/usbdisk, the second one `media/usbdisk1`, etc.

Mounted devices are shown as live information icons. For managed devices, the administrator can suppress the display of live information icons.

> **Note:**
>
> Due to security reasons, **Allow mass strorage devices** must be selected on the Hardware tab. Drive mapping for access to local resources must be defined in the relevant application definition. For Citrix ICA applications, see Citrix software defaults For RDP applications, see **Advanced application settings / RDP and VMware**

Printer tab

eLux® supports printing from local applications both to locally connected printers and to network printers. In addition, other systems or servers within the network can use a locally installed printer on a device running eLux. Next to the protocols LPR and TCP direct printing, proprietary protocols are also available.



In the Scout Console, in **Device configuration** > **Printer** > **New**, you can define and configure local printers with logical names.

Sharing printers

All printers defined in **Device configuration > Printer** can be shared with other systems via LPD within the network.

1. In **Device configuration > Printer**, select the **Print service activated** option.
2. Activate the Windows LPD service (Line Printer Daemon).

This option ensures that the print service is started at the device. All printers defined in the list can be used to print jobs from network devices.

The printers are controlled by the CUPS server.

Selecting a default printer

1. In the Scout Console, for the relevant OU or device, open **Advanced device configuration > Printer**.
2. In the **Default printer** list, select the printer that you want to be the default printer.

The list provides all defined printers for this element.

Citrix® auto-created printers

Citrix XenApp® provides automatic configuration of printers (**autocreated printers or dynamic printer mapping**). When the user logs on through a Citrix connection, an automatic printer definition is created on the Citrix server. The printer definition can only be used by the logged-on user and is deleted when the user logs off.

Citrix uses either the specified printer driver or, if not available, the universal Citrix printer driver, which is not tied to any specific device.

Configuring local printer for auto-creating on the device:

1. In **Device configuration > Printer**, specify one or more printers.
2. In the **Define Printer** dialog, in the **Name** box, enter the Microsoft Windows printer's name precisely as listed in the drivers list of the server. The name is case-sensitive.

When the user connects to the Citrix server, the automatically created device printers are shown in the printer settings.

If the specific driver is not installed on the application server or the name is not identical, the printer can not be created and the universal Citrix printer is used.

CITRIX UNIVERSAL™ PRINTING

The universal Citrix printer and various printer settings can be configured on the Citrix server, administrator rights provided.

For further information, see the **Citrix Product Documentation**.

TCP direct print

The print data can be received directly via TCP/IP and sent to the parallel port or USB port to the printer. The data are not modified before printing and there is no spooling of print jobs. TCP/IP handles the flow control.

CONFIGURING TCP DIRECT PRINT

1. In **Setup> Printer**, under **TCP direct print**, select the option **Enabled**.
2. Specify the relevant port number for the communication. The default port numbers are: 9101 for USB printers 9100 for parallel port printers

> **Note:**

> The specified ports are opened on the device.

> To print from a Windows session, for the printer port, choose a standard TCP/IP port. Specify the client IP address and the TCP/IP port selected in the previous step. Select for the protocol in Windows.

ThinPrint

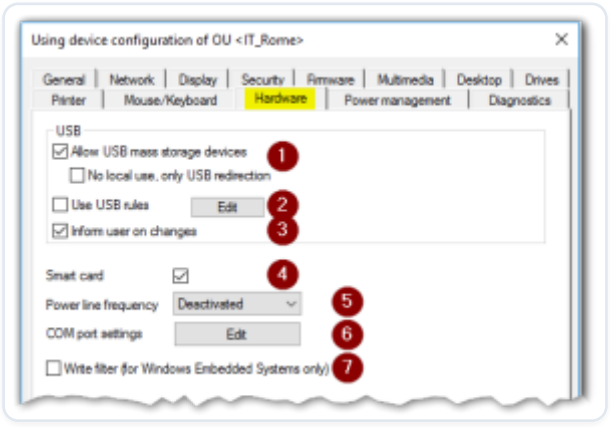
ThinPrint software from ThinPrint GmbH allows optimized network printing across various platforms. ThinPrint is a print protocol that, unlike TCP direct print, LPR or CUPS, allows bandwidth limitation. It is therefore recommended for networks with low bandwidth (WAN).

The software consists of a server component and a client component. The ThinPrint server processes the print data for the target printer and sends them to the device in compressed form. The ThinPrint client receives the print jobs from the server, decompresses and forwards them to the selected printer. ThinPrint server and client are connected via TCP/IP.

CONFIGURING THINPRINT

1. Install the ThinPrint client on the device.
2. Connect a printer.
3. In **Device configuration > Printer > New**, define the printer, and under **ThinPrint**, select the **Connect** option. Optionally, enter a class name of up to 7 characters.
4. If you use Windows CE clients, in **Device configuration > Printer** under **ThinPrint**, select the relevant protocol.
5. Configure the ThinPrint server. For further information, see the ThinPrint documentation on www.thinprint.com.

Hardware tab



1	Allows using connected USB mass storage devices. If the local use of USB devices via mount points is allowed, connected USB devices are shown as live information on the system bar. Select "No local use, only USB redirection" to restrict the use of USB mass storage devices to USB redirection within configured sessions on a backend. There are no mount points then provided locally on the eLux® device.
2	Restricts the use of USB mass storage devices according to defined rules. For example, you can restrict to devices with specific VID (Vendor ID) and/or PID (Product ID) such as an individual USB stick model. Moreover, USB rules can be applied to further USB device classes such as smart card readers. To define USB rules, click "Edit".
3	When a USB mass storage device is connected or disconnected, a live information message is displayed.
4	Enables smart card readers on a USB port.
5	Allows changing the refresh rate of webcams to 50 Hz or 60 Hz.
6	Set particular COM port settings such as speed, parity, stop bits.
7	Users are not allowed to store local data on their Windows Embedded device.

Note:

To enable users to use smart card readers, ensure to install the relevant middleware on the devices. `sc/interface` by Cryptovision is smart card middleware that integrates smart cards and other smart tokens into IT environments.sc/interface supports more than 90

> different smart card profiles.For further information, see the Cryptovision web page. To use **sc/interface**, the eLux package **Cryptovision sc/interface PKCS11** must be installed on the devices. This may require modifications of the image definition file on the web server via ELIAS.

- Defining rules for using USB devices
- Defining rules for USB redirection
- Safe removal of USB devices
- Enabling Bluetooth audio devices
- Webcams

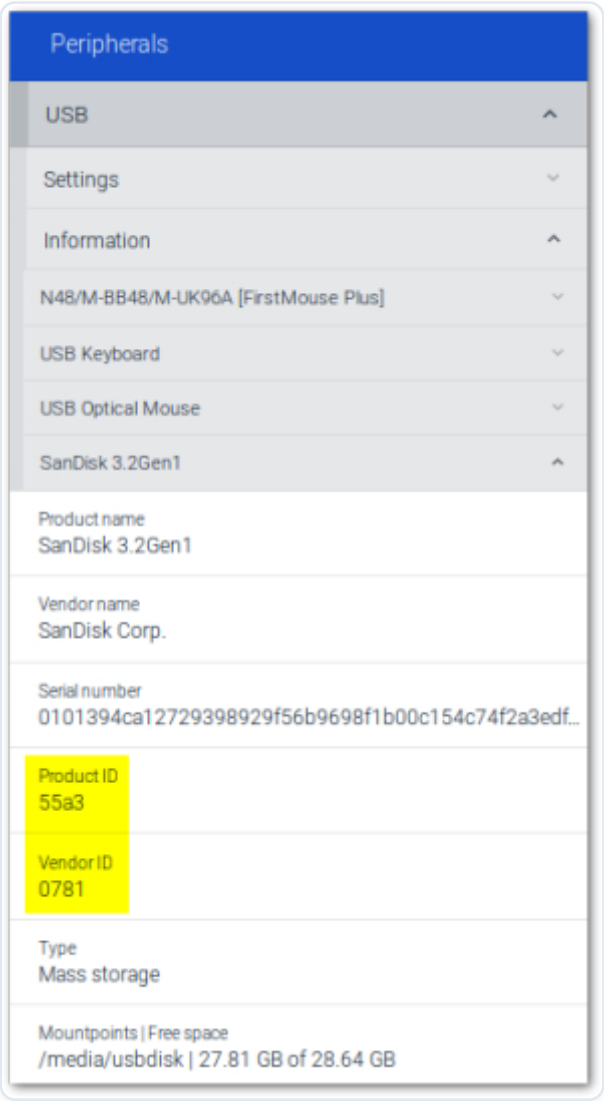
To display hardware information about the device itself or about connected (/en-us/unicon-elux-scout/2611/admin_guides/scout/interface_/interface_windows/interface_win_asset.html).

Defining rules for using USB devices

USB rules allow you to restrict the use of USB mass storage devices to certain models, for example.

1. For the relevant OU or device, open **Device configuration > Hardware > USB > Edit**.
2. In the list-field, select a set of predefined rules as template.
3. Double-click into the relevant line, or select a line and press F2.
4. Modify the rule by using the example rules below.

   The values of the manufacturer ID (VID) and product ID (PID) can be found on the device in the Config panel under (**Peripherals > USB > Information**):



5. Confirm with **OK**.

EXAMPLE RULES

RULE	CODE
Allow a specific USB mass storage device model only	`[ALLOW: VID=0781 PID=5151] # Allow a particular USB model (Example: SanDisk Cruzer Micro)`
Deny all devices of the class MASS STORAGE DEVICES	`[DENY: CLASS=08]`
Deny a specific smart card model only	`[DENY: VID=18a5 PID=0302] # Deny a particular smart card model (Example: Omnikey CardMan 3821)`
Allow all devices of the class SMARTCARD	`[ALLOW: CLASS=0B]`

RULE	CODE
Deny all printers, mass storage devices, smart card readers	`[DENY: CLASS=07] # Deny all devices of the class [PRINTERS], [DENY: CLASS=08] # Deny all devices of the class [MASS STORAGE DEVICES], [DENY: CLASS=0B] # Deny all devices of the class [SMARTCARD]`
Disable the microphone of a webcam	`[DENY: VID=045e PID=0810 CLASS=01] # Deny audio for the specified USB device`

The syntax of USB rules corresponds to the syntax of Citrix USB policy rules.

Important:

The USB rules affect all USB device classes including 03 HID (Human Interface Devices). If you deny the 03 HID class, the mouse and keyboard will be deactivated. For specific hardware configurations, you might encounter issues during the boot process of the device. We strongly recommend performing tests before using this option.

Defining rules for USB redirection

For the and 4.1 and later versions, you can define USB filtering rules for USB redirection of connected USB devices from eLux.

1. Type the required USB filtering rules into the appropriate configuration file:

APPLICATION	CONFIGURATION FILE	EXAMPLE RULES
Citrix	`/setup/ica/usb.conf`	`ALLOW: VID=0781 PID=5151`
\^	\^	`DENY: CLASS=08`
VMware	`/setup/elux/.vmware/default-config`	viewusb.ExcludeFamily = "storage"

APPLICATION	CONFIGURATION FILE	EXAMPLE RULES
\^	\^	viewusb.IncludeVidPid = "vid-0781_pid-5151"

2. To transfer the configuration files to the devices, use the Scout feature **Files configured for transfer**. For further information, see Files configured for transfer.

On the next restart of the relevant devices, the USB redirection rules become active.

> **Note:**
>
> The `/setup/ica/usb.conf` file is only required for the Citrix Virtual Apps and Desktops policy **Client USB device redirection rules**. If your environment uses **Client USB device redirection rules (Version 2)**, you do not need to create a `usb.conf` file.

Safe removal of USB devices

Any connected USB mass storage devices should always be removed by using the **Remove safely** feature to ensure that all data are saved.

Users on their device can right-click the USB icon (live information icon) on the system bar and then click **Remove safely**.

DEFINING A KEY COMBINATION FOR SAFE REMOVAL OF ALL USB DEVICES

In the Scout Console, you can define a key combination that allows the users to remove all connected USB mass storage devices safely in one step. The following instructions use the key combination alt+Windows+S as an example:

1. In the Scout Console, for the relevant devices, open **Advanced device configuration > Advanced file entries**.
2.

File	Section
Entry	Value
/setup/terminal.ini	Layout
UsbUnmountHotKey	`<Alt><Mod4><Hyper>s`

For further information, see Advanced file entries.

Enabling Bluetooth audio devices

You can allow the use of Bluetooth audio devices centrally from the console. Users can then search for devices in the eLux® Config panel and connect.

- In the , for the relevant devices, configure the following Advanced file entry:

File	/setup/terminal.ini
Section	Bluetooth
Entry	Enabled
Value	true

For further information, see Advanced file entries in the **Scout** guide.

For further information, see also Connecting Bluetooth audio devices in the **eLux** guide.

Webcams

Webcams on devices are listed under **USB > Information** even if they are built in. To allow users to preview a webcam image, specify an app that can be used to display it.[1[from eLux® RP 6 2107]]The app supports multiple cameras, both built-in and USB-connected cameras.

DEFINING AN APP FOR CAMERA PREVIEW

-- from --

> **Note:**

> In the **eLux Desktop Extensions** package, the feature package **Web camera preview** must be installed on the devices. This may require modifications of the image definition file on the web server via ELIAS.

1. Add a new application and, in the **Application properties**, select the application type [Defining custom applications].
2. Edit the following fields:

OPTION	DESCRIPTION
Name	Name for the application
Application	Custom
Parameter	bash cameraPreview

3. Confirm with **Apply** and **OK**.

When users launch the app, they select one of the connected cameras to get a preview of the camera image.If only one camera is available, the preview window is displayed when the app is started.

Note:

To disable the microphone of a webcam, define a USB rule.For further information, see Defining rules for using USB devices.

Diagnostics tab

The **Diagnostics** tab allows you to configure the **Device diagnostics**, which can either be requested by the Scout admin or sent from the device.

OPTION	DESCRIPTION
Debug level	If the **Debug level** is set to On (Enhanced debugging), by using the **Device diagnostics** feature you run predefined commands on the device and retrieve a set of configuration and log files to a greater extent than without enhanced debugging.
	If you require technical support from Unicon™, switch on enhanced debugging before you perform **Device diagnostics**.

Note:

Make sure to switch off the after having performed device diagnostics. Otherwise, you risk exceeding the memory capacity of the device.

Send files to	FQDN of a [Scout Server] or URL of an FTP server to be used as the destination server to which the devices send their diagnostic files. By default, the diagnostic files are transferred to the Scout Server the device is connected to. This option allows you to specify, for example, a specific [Scout Server] or FTP server that will receive the diagnostic files for all devices regardless of their assigned Scout Server. The specified destination server is also active when diagnostic files are sent from eLux.
Enhanced logging for smart card support	Creates an additional log file [/tmp/PCSCDlog.txt] when PCSC Lite is used

Device diagnostics is performed by using a Scout command. For further information, see Device diagnostics.

Power management tab

Especially when using mobile devices, you will need options for power management. By using profiles, on the **Power management** tab, you can pre-define power management settings for your computer. These settings become active when you or the system enable the relevant profile:

- High performance: Favors performance, but may use more energy
- Power saver: Saves energy by reducing computer performance and screen brightness

You can either explicitly activate one of the power management profiles or, for mobile devices, you can let the system choose by selecting the Auto option: If the device is plugged in, the profile **High performance** will be active. If the device is on battery power, the profile **Power saver** is activated.

For both profiles, you can additionally distinguish between working and non-working hours. You can further optimize energy consumption by making more rigorous use of energy saving options outside working hours. To do so, you are required to specify your working hours first. Once you have defined working hours, based on these, the system automatically switches between the **Working hours** profile and the **Default** (Non-working hours) profile. Important: If you choose not to

define working hours, the system will use the **Default** profile settings, regardless of the day of the week and time.

Setting a power management profile

- On the **Power management** tab, from the [**Active power management profile** list, select a profile or the Auto option.

The settings defined in the profile become active. The Auto option activates the **High performance** profile if the device is plugged in, and the Power saver profile if the device is on battery power.

Profile-independent options

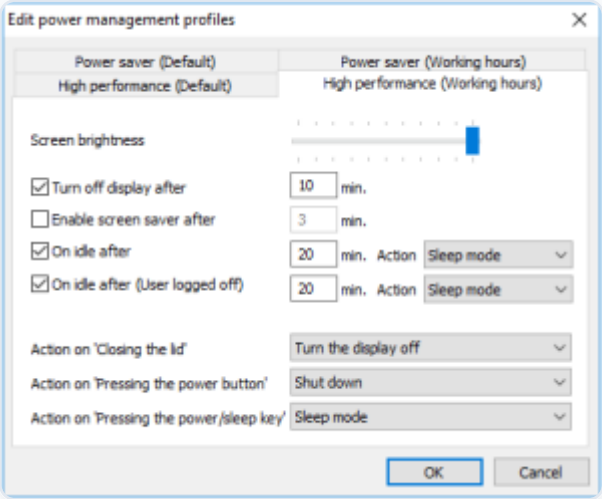
OPTION	DESCRIPTION
Low battery notification	For devices without power supply: Battery status in percent, from which the user is notified
Log off user before going to sleep mode	Users must log on after the computer wakes from sleep mode.

- Configuring power management profiles
- Defining working hours
- Sleep mode (Suspend)

Configuring power management profiles

> **Note:**

> If you define your working hours, you can also specify which energy-saving options you want to apply outside of working hours. As long as you do not define working hours, the energy-saving options apply non-stop (24/7).



1. Next to **Power management profiles**, click **Edit** and then select the tab for the required profile. To define profiles for non-working hours, first define the working hours.
2. For each profile you want to use, edit the following fields:

OPTION	DESCRIPTION
Screen brightness	Screen brightness in percent for the selected profile. Move mouse pointer over the slider to display the percentage.
Turn the display off after	Determines whether, after a specified number of minutes, the display is turned off when the user is not using the device (idle state).
Enable screen saver after	Determines whether, after a specified number of minutes, the screen saver is enabled when the user is not using the device (idle state).
On idle after – Action	Determines whether, after a specified number of minutes, the selected action is performed when the user is not using the device (idle state): - Shut down, - Sleep mode, - Restart, - Log off
On idle – Action (User logged off)	When the user is logged off: Determines whether, after a specified number of minutes, the selected action is performed when the user is not using the device (idle state).

OPTION	DESCRIPTION
Action on 'Closing the lid'	Action that is performed when the user is closing the lid: - No action, - Turn the display off, - Shut down, - Sleep mode
Action on 'Pressing the power button'	Action that is performed when the user presses the power button: - No action, - Turn the display off, - Shut down, - Sleep mode
Action on 'Pressing the Power/Sleep key'	Action that is performed when the user is pressing the Power/Sleep key on their keyboard (requires a suitable keyboard): - No action, - Shut down, - Sleep mode

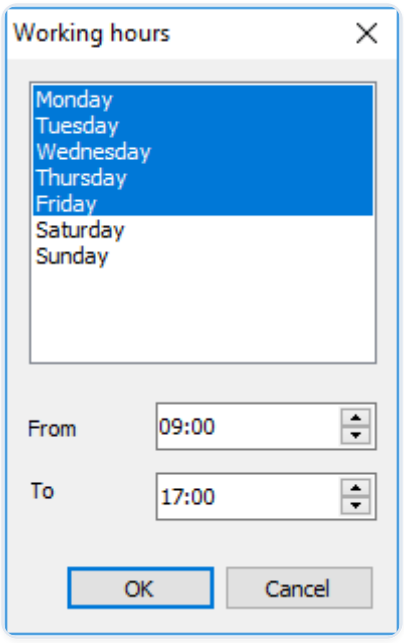
3. Confirm with **OK**.

> **Note:**
>
> The sleep mode corresponds to **Suspend to RAM (/en-us/unicon-elux-scout/2611/admin_guides/scout/setup/power_man/suspends3.html).

Defining working hours

REQUIRES
User right **Define working hours**

1. On the **Power management** tab, next to **Working hours** click **Edit**.
2. Select all weekdays that are working days and that are relevant to working times. To select multiple entries, press Shift or Ctrl.
3. Select the earliest time for the start of work (**From**). This time refers to all defined working days.
4. Select the latest time for the end of work (**To**). This time refers to all defined working days.
5. Confirm with **OK**.



Once you have defined working hours, the **High performance** and **Power saver** profiles are divided into sub-profiles for **Working hours** and **Default** (non-working hours). Switching between the two sub-profiles is based on the working times you have defined.

Sleep mode (Suspend)

The sleep mode corresponds to **Suspend to RAM (S3)**.

To put their device into sleep mode, users click a button in the extended Command panel. The sleep mode can also be activated by the system if the power management options are configured accordingly.

When the device goes into sleep mode, the eLux® user remains logged on by default. After reactivating the device, users can continue working right away. This assumes that the backend is configured accordingly: When an application is disconnected, the user must not be logged off automatically.

If you want users to log on and authenticate again after the device wakes from sleep mode, select **Log off user before sleep mode** in the **Device configuration > Power management**. The user is then logged off and when the device wakes from sleep mode is given the eLux logon dialog.

> **Note:**
>
> In the Scout Console, the **Properties** window shows the status of a suspended device as Switched off (Suspended).

Troubleshooting device configuration

The solutions provided below refer to the Scout Console in the first place.

ERROR / PROBLEM	REASON	SOLUTION
When you use USB multimedia devices such as headsets or webcams, the screen freezes or the window cannot be focused.	The USB operating elements register themselves as keyboard or mouse devices in the system.	Prevent the registration as input devices by defining a `terminal.ini` entry. The basic functionality of the operating elements is not affected. For further information, see Preventing registration of USB multimedia components.
Multimedia USB devices, connected via DisplayPort to eLux RP 5 devices with an AMD processor, do not play back sound.	Sound reproduction via DisplayPort is disabled.	Enable sound reproduction by defining a `terminal.ini` entry. To do so, use the Scout feature Advanced file entries: **File:** `/setup/terminal.ini`, **Section:** `Screen` , **Entry:** `Radeon.Audio` , **Value:** `true` , Alternatively, use a separate audio cable.
Monitor via DisplayPort with AMD GPU: After changing to lower resolution the monitor brings an "Out of range" error message.	The resolution on this monitor interferes with the configured sound reproduction via DisplayPort.	Disable sound reproduction via DisplayPort. This will fix the monitor error. To do so, use the Scout feature Advanced file entries: **File:** `/setup/terminal.ini` , **Section:** `Screen` ,**Entry:** `Radeon.Audio` , **Value:** `false`
When you use a touch screen, the location of a fingertip touch is not recognized precisely.	The monitor has not been calibrated precisely enough.	To calibrate the monitor, configure a custom application by using the parameter `calibrator`. Then start the application.

ERROR / PROBLEM	REASON	SOLUTION
Only eLux RP 5.7.x: In dual monitor mode, if the second monitor is configured to vertical, the desktop icons are not displayed (correctly).	For some resolutions, the desktop icons on the primary monitor cannot be displayed when the second monitor is vertically aligned and the lower screen area is referenced.	For eLux RP 5.7.3000 and later versions: Use a new parameter to configure the vertical alignment to the upper screen area (top). To do so, use the Scout feature Advanced file entries: **File:** `/setup/terminal.ini` , **Section:** `Screen` , **Entry:** `VerticalAlignment` , **Value:** `top` , The default value is `bottom` .
Display/general graphics issues	The feature package for hardware acceleration `HwVideoAccDrivers` is not installed.	Activate the `HwVideoAccDrivers` FPM within the `XOrg` package in the IDF.
	Hardware acceleration (installed with the `HwVideoAccDrivers` FPM) is not supported by the device and causes problems.	To exclude individual device types from hardware acceleration, create a blacklist that is transferred and locally saved to the devices by using the Scout feature Files configured for transfer: **File:** `/setup/hwaccBlacklist` , In the text file `hwaccBlacklist` , list the relevant device types, one per line. The name of the device type must be identical to the string that is shown in the Scout Console, in the Properties window under `Asset > Hardware information > Type` . ,**Example:** , `FUTRO S920` , `D3314-B1` , `HP t620 Dual Core TC` . For all device types listed in the

ERROR / PROBLEM	REASON	SOLUTION
		blacklist, hardware acceleration is disabled.
AD logon to eLux RP 6.x does not work.	Port 389 is configured for the authentication server.	Do not define a particular port for the authentication server.

> **Note:**
>
> After the `terminal.ini` file has been updated on the client, another client restart might be required to enable the new setting.

Preventing registration of USB multimedia components

The registration of USB devices as input devices can be prevented by defining a terminal.ini entry. To do so, use the **Advanced file entries** feature of the Scout Console.

1. In the Scout Console, for the relevant devices, open **Advanced options > Advanced file entries**.
2. Define the following entry

File	/setup/terminal.ini
Section	Xorg
Entry	IgnoreUsbInput
Value	VendorID_1:ProdID_1,VendorID_2:ProdID_2
(VendorID)	Example: 0b0e:034c,047f:c01e
or	Important: Only use lowercase letters for hexadecimal values!
	You can replace individual characters by the wildcard character ?.
	Example: 0b0e:???? filters all products of a specific vendor.

Value	Alternatively, you can filter by vendor name
(VendorName)	Example: Jabra,Netcom
	- White spaces or slashes in the vendor name must be replaced by an underscore _.
	- The vendor name can be entered as a sub-string.
	Example: Netcom finds GN Netcom and GN Netcom A/S.
	- Vendor names can be OR-linked
	Example: Jabra|Sennheiser
	Note: To identify vendor names or IDs, use the following command
	udevadm info \--export-db | grep -Ew \"(NAME|ID_VENDOR)\"

For further information, see Advanced file entries.

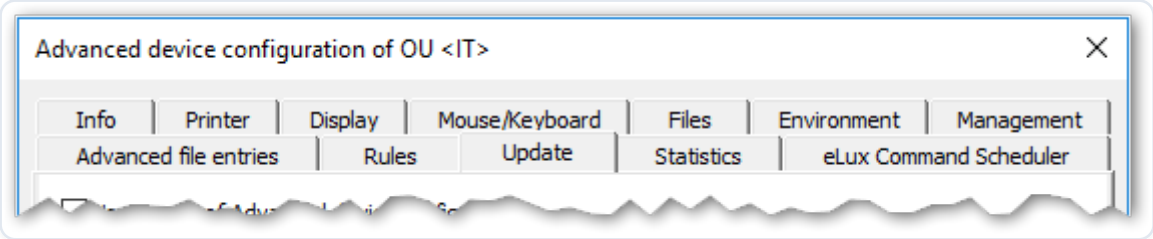
> **Note:**

> After the terminal.ini file has been updated on the client, another client restart might be required to enable the new setting.

Advanced device configuration and Advanced options

The device configuration defined globally in **Options** > **Base device configuration** or for individual OUs or devices in their **Device configuration** can be extended as follows:

- Override applied options and add further options for individual devices or OUs by using the **Advanced device configuration**

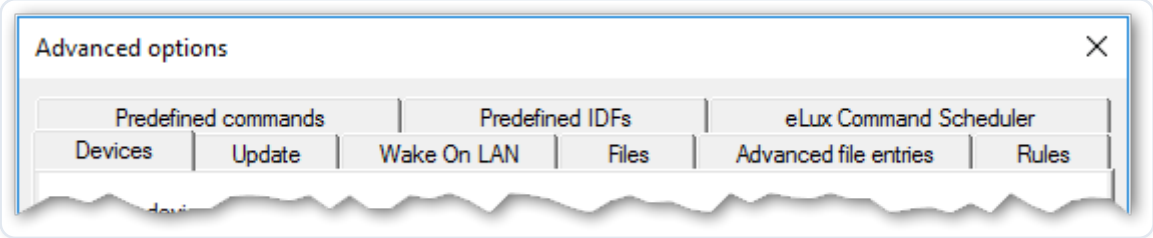


The **Advanced device configuration** is applied on OU or device level.

- Open the relevant context menu and click **Advanced device configuration...**

Inheritance is also used for Advanced device configuration. By default, the option **Use parent of Advanced device configuration** is selected. You can, however, clear this option on some of the tabs.

- Apply further options to all devices by using the **Advanced options**



The **Advanced options** are applied to all devices.

- On the **Scout Console** menu, click **Options** > **Advanced options**.

Some functions such as **Files** (configured for transfer), **Advanced file entries** and **Update** options can be found in both dialogs. Use **Advanced device configuration** to apply the function to individual devices. Use **Advanced options** to apply the function to all devices.

Devices

only globally available for all devices **Options > Advanced options**

Discover devices

OPTION	DESCRIPTION
Maximum ping-time (milliseconds)	Maximum response time in milliseconds by which devices should respond to a `ping` command.
Maximum discover time (seconds)	Total time for searching devices for Discovery. After the indicated time has expired, Discovery is stopped.

Field update

OPTION	DESCRIPTION
Retain local configuration (unlocked fields)	Editable fields of the local device configuration are free for individual user configuration and will not be overridden by Scout. During synchronization of configuration data from the Scout Server to the device, only the values of the locked fields will be overridden.
	Which options are allowed to the user for editing (unlocked fields) is determined by the administrator under **Device configuration > Security > User rights**.
	The local user configuration values of unlocked fields can additionally be retained during a factory reset of the device.
	For further information, see Supporting local configuration.

Note:

If users have set defective configuration data, you can, however, override unlocked fields and set a flag for the relevant device in the Scout Console to reload all configuration data.For further information, see Supporting local configuration

New devices

Default OU	OU new devices are assigned to, by default
Assign OU depending on OU filter	Activates the OU filter for new devices
	Click the ... button to configure the OU filter.The OU filter has priority over other methods but can be ignored for individual devices.For further information, see OU filter.

Lock config transfer for new devices	Newly added devices are not synchronized with the server's device configuration
Allow dynamic change of OUs	Allows dynamic assignment of devices via DHCP
Accept only known devices	The Scout Server accepts only devices with registered MAC addresses. For further information, see Reserving device profiles.

Device name

Use the host name of the device	The device name is the client host name and cannot be changed in the Scout Console permanently.
To avoid duplicate names, change name of existing entry	When a new device with an already existing name is added, the name of the existing device instead of the name of the new device is changed.
Name template	Name template for new devices
	Can be overridden for particular OUs (**Advanced device configuration > Management**)
Apply name template only on new devices	Name templates are not applied when you move or relocate devices.

For further information, see Device names.

Update/Delivery

Not available for individual devices

OPTION	DESCRIPTION
Maximum number of parallel updates or software deliveries	Restriction for performance reasons

OPTION	DESCRIPTION
Maximum time to connect	Time for setting up the connection before the next device is accessed

> **Note:**
>
> The optimum values depend on the individual system.

Management

only available for individual devices and OUs

OPTION	DESCRIPTION
Note	Free text field for internal comments
	Can be shown in the **Properties** window
Visibility	The visibility refers to the list of OUs that new devices request when they initially connect to the [Scout Server] to select an OU.
only for OUs	Hidden OUs are not shown in the list.
	Visible OUs can be selected in the list but can be protected by password.
New devices	When new devices register automatically, they can be subject to a predefined name template.
only for OUs	Device names are configured globally in the **Advanced options** on the **Devices** tab. Please note the parameter **"Apply name templates only on new devices"**. For further information, see Devices.
Ignore OU filter	If the OU filter is active (**Advanced options > Devices**), new devices are assigned to OUs due to the defined criteria.

OPTION	DESCRIPTION
only for individual devices	Individual devices can be excluded from the OU filter.
	By moving devices within the tree structure by a drag-and-drop operation, for the relevant devices, the option is selected automatically.

Password-protected and hidden OUs

From UniconOS Management 2611, Scout Server itself checks the OU password when a new device joins a password-protected or hidden OU, and when a known device moves itself into such an OU. A hidden OU counts as protected, also if it has no password. Devices that are already in the OU are not checked when they contact Scout.

By default, Scout Server places a new device that doesn't send the password in the default OU. Some devices never send it: devices placed by the DHCP option 223, by a GroupID in the image or by Scout Cloud Gateway, and devices that users set up with the setup wizard of UniconOS 2611 or earlier. To place such devices in a protected OU, use an OU filter rule, see OU filter. For the setting that decides about devices without the password, see Preferences.

Wake On LAN

only globally available for all devices (**Options > Advanced options**)

Wake On LAN is a feature supported by Scout that helps you start turned-off devices.

REQUIRES
Wake On LAN is supported by the device's producer and enabled in the device's UEFI/BIOS.
Another device in the same OU is already online.

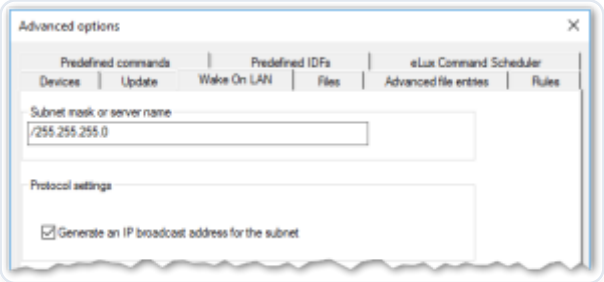
The Scout Server sends a so-called magic packet that is identified by the network adapter of the turned-off devices. The devices are woken via port 67.

The magic packet for Wake On LAN is sent as a broadcast (UDP, eLux port 20000 incoming/outgoing) within the current network subnet and cannot operate across the entire network. To wake up devices in remote subnets, subnet directed broadcasts are required.

Subnet directed broadcasts

Subnet directed broadcasts can directly address the device's subnet to be woken up via IP. The IP broadcast address of the relevant subnet is determined from the IP address of the device and the configured subnet mask. The magic packet for Wake On LAN is broadcasted (UDP) only within the addressed subnet.

An IP broadcast address for the subnet must be configured once in the Advanced options.



OPTION	DESCRIPTION
Server name	Subnet mask for subnet directed broadcasts (for earlier versions: IP address of Wake On LAN server -- option also available in the Advanced device configuration)
Generate an IP broadcast address for the subnet	The packet is sent to the relevant subnet (dedicated subnet). Requires a subnet address in the Server name field using the format / 255.255.255.0 (Note the leading slash).
(only globally available in Advanced options)	Example: To wake up a device with IP address 192.168.10.44, enter /255.255.255.0 in the Server name field. This causes Scout to generate the IP broadcast address 192.168.10.255 for the subnet. By default, this option is disabled.

VPN

only available for individual devices

> **Note:**

> One or more VPN profiles can be defined for entire OUs in **Device configuration > Network** .[1[from Scout Enterprise Management Suite15 2101]]

Scout supports the following VPN (Virtual Private Network) clients for secure communication:

- Cisco AnyConnect / 2[from eLux RP 6 2302.1000]
- OpenVPN

Depending on the VPN client used, the devices must have a configuration file. You can modify the configuration file by using the Scout feature Advanced file entries.

- Configuring Cisco AnyConnect / Cisco Secure Client
- Configuring OpenVPN

Configuring Cisco AnyConnect / Cisco Secure Client

Cisco renamed its VPN **AnyConnect** Client from version 5 to **Cisco Secure Client**. The VPN technology remains the same, and the protocol is still called AnyConnect. Unicon ships the Cisco Secure Client for the first time with eLux RP 6 2302.1000, in parallel with the well-known AnyConnect Client. Their configuration for eLux is the same for both packages.

The following instructions refer to individual devices. To define VPN profiles for entire OUs, use **Device configuration > Network > VPN**. For further information, see Defining a VPN profile.

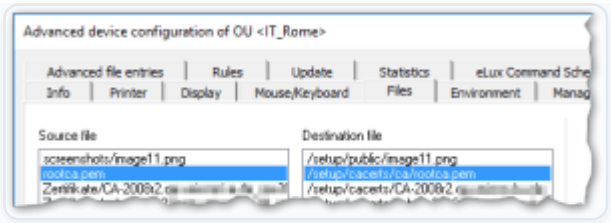
Note:

The eLux® package **Cisco AnyConnect** or **Cisco Secure Client** must be installed on the devices. This may require modifications of the image definition file on the web server via ELIAS.

1. Transfer the root certificate to the devices to /setup/cacerts. If you use the Scout feature Files configured for transfer

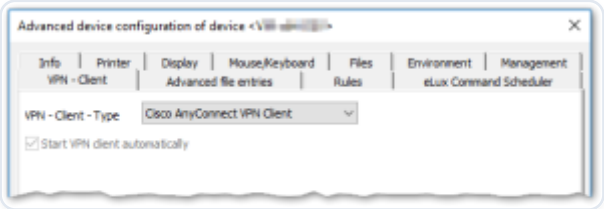
   Note:

   Use the `.pem` format for the certificate.



> **Note:**
>
> The certificates that are transferred from the VPN server are stored in /setup/cacerts/client.

2. In the Scout Console, for the relevant device, open **Advanced device configuration > VPN client**. Then, in the list-field, select `Cisco AnyConnect VPN Client`. Use this option also for the Cisco Secure Client.



3. Restart the device.The device might require one more restart to activate the VPN configuration data locally.

The device connects via AnyConnect on the next restart. Users have the active VPN connection as a live information on their system bar.

CONFIGURATION FILE

As an option, you can create an AnyConnect or Cisco Secure Client configuration file or copy one from a reference device, and then transfer the file to `/setup/elux/.cisco/profile/` via the Scout feature Files configured for transfer. In the configuration file, you can specify your back-end server address.

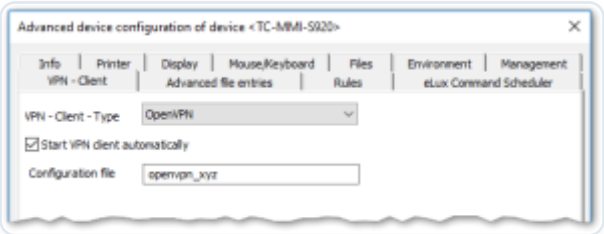
Configuring OpenVPN

The following instructions refer to individual devices. To define VPN profiles for entire OUs, use **Device configuration > Network > VPN**. For further information, see Defining a VPN profile.

> **Note:**

> **OpenVPN** is an integral part of the eLux® operating system. There is no need to have a separate package installed on the devices.

1. Transfer the .ovpn configuration file and certificates to the devices to /setup/openvpn. If you use the Scout feature **Files configured for transfer**, specify the source and destination file **with** file name extension and the destination path /setup/openvpn. If you use a USB stick, unzip the .zip file to the client directory /setup/openvpn.
2. In the Scout Console, for the relevant device, open **Advanced device configuration > VPN client**.Then, in the list-field, select .



3. Select the option **Start VPN client automatically**.
4. In the **Configuration** field, enter the name of the OpenVPN configuration file **without** the file name extension `.ovpn`.

On the next restart of the device, the VPN configuration file is transferred to the device and activated with another restart.The OpenVPN logon dialog is displayed and the user can connect.

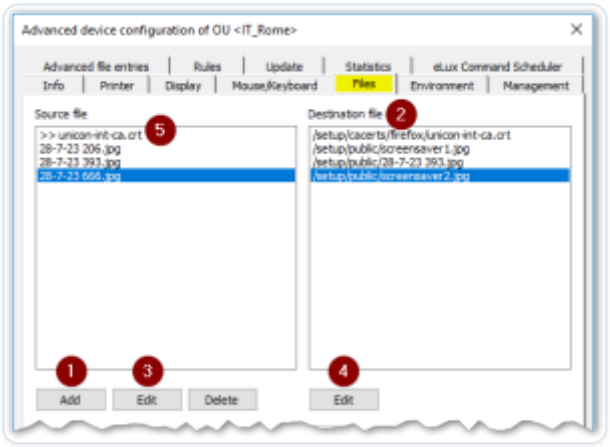
> **Note:**

> The OpenVPN client can only be used with a valid .ovpn configuration file under eLux. Ensure that the configuration entries are correct.eLux does not accept the setting of an additional default route via the .ovpn configuration file. Some firewall vendors offer different configuration files for different operating systems, with and without a default route. For example, the Sophos configuration file for Android/iOS does not include this default route and can be used for eLux.

Files configured for transfer

This feature helps you transfer files to the devices. The defined files are transferred on the next device restart. You can apply a file transfer to all devices, to individual devices or to OUs.

The source files are imported to the Scout database and therefore are included in an SQL database backup.



Example: Picture files are copied to the devices as screen savers.

1	The source files are selected from the file system via **Add**.
2	For each source file, a destination file on the device is created
	Specify the relevant destination directory and optionally a different file name.
3	Source files of the current level can be edited.
4	For entries of the current level, you can edit the destination file properties.
5	Entries from a parent OU or the global file list are identified by a **>>**.
	You cannot edit these source files. With the corresponding object right, however, you can view them,[1[[ ] from Scout 15 2101]] see below.

Defining files for transfer

1. To configure a file transfer to all devices (global file list), click **Options > Advanced Options** On top level, you can also define destination file templates for subordinate levels.

   To configure a file transfer to the devices of an individual OU or to an individual device (individual file list), on the context menu of the relevant OU or device, click **Advanced device configuration**

> **Note:**
>
> Individual file lists have precedence over global file lists.

2. On the **Files** tab, click **Add**.
3. In the **Add file entry** dialog, to select the source file from the file system, click the … button and then select a file.

   A new entry for the **Source file** and **Destination file** lists is created.
4. Under **Destination file**, if required, modify the directory and file name of the destination file on the device. The destination file name may differ from the one of the source file.

   If configured, alternatively use a destination file template from the **Template** list field. The path and name of a destination file from a template cannot be changed later on.
5. Confirm with **OK**

Source and destination are defined. The files are transferred on the next device restart. The files will not be reloaded unless you modify the file configuration.

Re-using imported source files

You can re-use source files that you have imported for one OU for the file transfer in other OUs.

1. In the Advanced device configuration of the source OU, on the **Files** tab, right-click the relevant entry in the **Source file** list.
2. To copy the entry to the Clipboard, from the context menu, choose **Copy**.
3. Open the Advanced device configuration of the target OU.
4. On the **Files** tab, right-click the **Source file** list. On the context menu, click **Paste**.

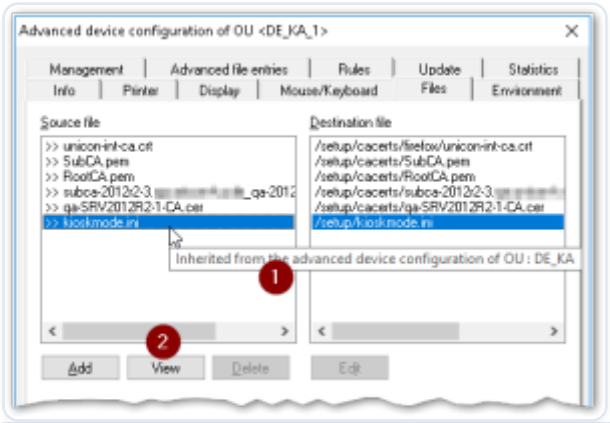
Removing transferred files

- To remove a transferred file from a device, on the **Files** tab, on the relevant level, select the file and click .[2[from eLux® RP 6 2209]]

> **Note:**
>
> To remove all transferred files on all levels, alternatively perform a factory reset. Also use a factory reset for earlier eLux versions.

Source files from parent OUs or the top level

The **Files** dialog also displays entries from parent OUs or the global file list. These inherited source files are identified by a **>>** character and are protected from access by default.



1	To show the origin of a parent entry, move the mouse pointer over the entry.
2	You cannot edit source files of parent entries.
	But, if the file privacy object right has been disabled, you can view the content of a source file. [from Scout 15 2101]
	To disable the object right, for the required OU and for the required administrator, open the object rights, and then navigate to **Edit properties > Files > Inherited file privacy**.
	For further information, see Changing object rights.

Advanced file entries

The **Advanced file entries** tab allows you to set parameters in `.ini` files that cannot be set by using the graphical user interface. For example, you can define additional layout parameters.

Advanced ini

1	Entries from a parent OU or the global **Advanced options** are shown with a **>>**. To

	show the origin, move the mouse pointer over the relevant entry.
2	For a new entry, first select the relevant file or enter the path and filename.
3	For the new entry, then specify the section, key and value.
4	Click **Add** to add the new entry.
5	Click **Replace** to replace the selected entry in the list by the new entry.

The following applies to the `.ini` files:

- `.ini` files contain at least one section. Each section contains zero or more keywords. The keywords contain zero or more values.
- Each section is headed by a symbolic name enclosed in square brackets.
- Each keyword and its value are in one line. Keyword and value are separated by an equal sign (=). A keyword can have more than one value.
- If a keyword is used more than once in the same section, the last occurrence has precedence.
- Adding individual file entries
- Changing values of individual file entries
- Deleting individual file entries
- Deleting complete sections

Adding individual file entries

1. In the Scout Console, click **Options > Advanced Options**. Or: For the relevant OU, open the context menu and click **Advanced device configuration**
2. Click the **Advanced file entries** tab.

3. Below the list, edit the following fields (1):

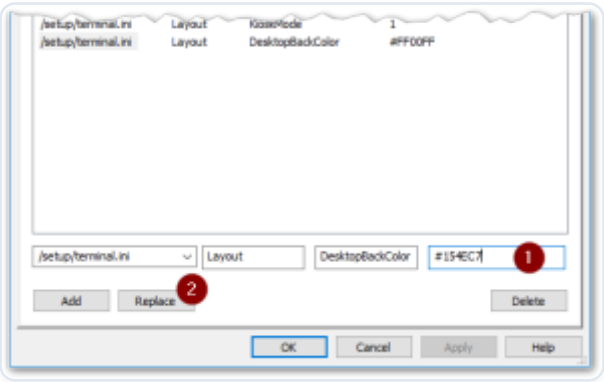
OPTION	DESCRIPTION
File	Enter the full path including file name or select from the list:
	Terminal: `/setup/terminal.ini`
	Citrix ICA: `/setup/ica/wfclient.ini` and `/setup/ica/appsrv.ini`
	Cisco VPN: `/setup/ciscovpn/sample.pcf`
Section	Section heading without brackets
Key	Keyword
Value	Value you want to assign to the keyword
	Blanks, hyphens and multiple values are allowed.
	Example: `valueA,valueB,valueC;comment`

4. Click **Add** (2).
5. Confirm with **Apply** and **OK**.

The new entry is written to the .ini file on the next device restart.

Changing values of individual file entries

1. In the Advanced device configuration > Advanced file entries, select the entry whose value you want to change.



2. Below, in the **Value** box, replace the current value.

3. Click **Replace**.

The new values are written to the .ini file on the next device restart.

Deleting individual file entries

1. In **Advanced device configuration > Advanced file entries**, define a new entry:
2. Enter **File**, **Section** and **Key** of the relevant file entry, but leave the **Value** box empty (1).

3. Click **Add**. (2)

The 'empty' file entry overrides previous instructions. The file entry is deleted from the relevant section on the next device restart.

> **Note:**
>
> If you use the Delete button to delete a selected row from the list, then Scout will no longer update the respective entry. However, the entry with its value remains.

Deleting complete sections

1. In the **Advanced device configuration > Advanced file entries**, define a new entry: Specify the **File** and **Section** of the relevant file entry, but leave the **Key** and **Value** fields empty.
2. Click **Add**.

The 'empty' section overrides previous instructions. The section is deleted from the file on the next device restart even if it contains file entries.

Rules

Using this feature helps you define rules which can be executed when closing the last application or using Scout for the first time.

OPTION	DESCRIPTION
After terminating the last application execute the following action	Choose between the options of the list-field

OPTION	DESCRIPTION
	For OUs and devices, the Use parent action option is set by default to enable the rules defined for a higher OU level.
Display a message on the device for `<x>` seconds	Enter a time period in seconds to inform the user
After first management contact execute the following action	Select Update the device to ensure that the new devices are all up-to-date.

Special case kiosk mode

If you use the browser in kiosk mode to access Citrix applications, the option **After terminating the last application execute the following action** does not work. In this case, you can set an entry in the `terminal.ini` file to determine an action that is performed after the last Citrix application is closed.

- by using the Scout feature Adding individual file entries:

File	/setup/terminal.ini
Section	Global
Entry	ActionAfterLastWfica
Value	
	0 - Use parent action
	1 - Restart
	2 - Shutdown
	4 - Logoff
	8 - Lock
	16 - VPN disconnected

Environment variables

only available for individual devices and OUs

Environment variables can be used locally on the device. They contain strings.

Defining environment variables

1. In the **Advanced device configuration**, click the **Environment** tab.

   Previously defined variables are shown in the list. Entries from a parent OU are shown with a **>>**.

2. Click **New**.
3. Enter the required variable using the format: `Variable name=value` and confirm with **OK**.

   The new variable is shown in the list.

4. To encrypt the value of the variable, right-click the variable. Then from the context menu, choose **Encrypt value**.

> **Note:**
>
> When you apply the defined variables later on, make sure you prefix the variable with a dollar sign: <Variable name\>. For further information, see also Where to apply user variables.

TPM 2.0 support

A TPM 2.0 chip built into a device can be used for basic security functions:

- Encryption of the setup partition and system partition

The setup partition on a device's flash memory contains the device configuration, application definitions, and certificate store. The system partition holds the software packages of the firmware.

In order to protect the system from manipulation, in addition to encryption, the disk is sealed with security measurements.

- Store the private key of a SCEP client certificate inside the TPM 2.0 module

To store the key inside the TPM 2.0 module, a `scep.ini` entry is required. For further information, see Certificates for SCEP in the `SCEP` guide.

> **Note:**
>
> If you want to use TPM 2.0 via WLAN, note the special parameters in the configuration file `wpa.conf`. For further information, see Configuring WPA supplicant in the **IEEE 802.1X** short guide.

Requirements for disk encryption

- The devices are provided with a TPM 2.0 module.
- The devices are started in UEFI mode.

Disk encryption via TPM 2.0

If the device-side requirements are met, encryption can be enabled using two different mechanisms:

- Via the configuration parameter **DiskEncryption**
- Via the feature package **Partition encryption** installed with the image

If you install the BaseOS package on the **devices** with the feature package **Partition encryption** enabled, the system will automatically be encrypted. The parameter **DiskEncryption** is then ignored.

The feature package **Partition encryption** is enabled by default.

To encrypt the disk, the partitions must first be formatted. Therefore - as soon as the encryption is activated - a firmware update with previous formatting for the relevant devices is forced.

Encrypting the disk via parameter

1. In the Scout Console, for the relevant devices, open **Advanced device configuration > Advanced file entries**.
2. Define the following entry:

File	`/setup/terminal.ini`
Section	`Security`
Entry	`DiskEncryption`

Value	`true` : Default Value `false`

For further information, see Advanced file entries.

For the relevant devices, a firmware update is forced with previous disk formatting.

The configuration parameter has no effect on devices without TPM 2.0.

> **Note:**
>
> You can find information on whether the disk of the device is encrypted in the **Properties** window.
>
> When new devices with TPM 2.0 chip are on-boarded to the Scout infrastructure and the destination OU is configured with DiskEncryption, it is ensured that the configuration data stored in the is only saved locally on the device after the setup partition has been encrypted.

Update from earlier versions with disk encryption

Updates with disk encryption can only be performed from eLux RP 6.x. Upgrades from eLux RP 5 are not supported.

If you enable encryption when updating to a current eLux® RP 6 version, one more update may be required on the next device restart. This is due to the partition formatting that is required for encryption.

Error handling

If a device fulfills the above-mentioned requirements for encryption and disk encryption still fails during the update, the setup partition will be partially cleaned like it is for a factory reset, without deleting the Scout Server address. The device status in the Scout Console is then displayed with a yellow icon (initialization).

Resetting the disk encryption

REQUIRES
The feature package [Partition encryption] must be uninstalled on the relevant devices. This requires modifying the image definition file on the web server via ELIAS.

- Set the advanced file entry `DiskEncryption` to the value `false`.

or

- Perform a factory reset for the devices. To do so, use the **Remote factory reset** command with the option **Delete Scout Server address on the device**.

During the restart of the relevant devices, the disk is decrypted. That is why the start up process takes longer.

Defining applications

The devices can be supplied with the following types of applications:

- Applications providing access to back-end systems
- Local applications

The definition of applications and the installation of the related software are independent of each other. Defining applications means to configure the applications provided for the users. Additionally, to enable the users to operate the applications, the relevant software packages must be installed on the device via IDF configuration. For further information, see Creating an image in the **ELIAS 18** guide.

> **Note:**

> The term **applications** refers to application definitions. The term **software** refers to the required software packages.

> Applications can be inherited from the top of the organization structure to subordinate OUs. The lowest level to define an application is an OU, the highest level is the root level.

- General
- Connecting to a Citrix farm
- RDP
- Virtual Desktop
- Browser
- Local and user-defined applications
- Emulation

- Applications in kiosk mode
- Local web sites
- Troubleshooting application definition
- Third party software

General

- Adding applications
- Editing application properties
- Defining free application parameters
- Parent applications
- Software defaults
- Defining application templates
- Uploading applications from a device to Scout
- Defining application icons
- Assigning custom application icons
- Defining a service app
- Limiting applications to one logon domain

> **Note:**

> You can define actions to be performed after the last application has been closed. For further information, see Rules.

> Additionally note that application definitions can be

- copied from one OU to another
- exported from one OU and imported to another OU (context menu > **Edit**).

Adding applications

1. In the tree view, right-click the **Applications** icon of the relevant OU.
2. On the context menu, click **Add**.

The **Application Properties** dialog opens. This dialog provides several tabs, each of them relating to a particular application type.

The following options of the **Application Properties** are available for most application types:

OPTION	DESCRIPTION
Name	Name of the application shown in the [Scout Console]
	Important: Applications are identified by their name. Make sure to use a unique name for them.
Display name (optional)	Name of the application shown on the device (control panel, start menu)
Sorting ID	Specifies the sorting order for applications pinned to the system bar
	[1] sorts alphabetically (default)
Server	Name of the server the application connects to
Login	The user is automatically logged on to the terminal server by using predefined credentials (username, password, domain).
Pass-through logon	The values of the local user variables `$ELUXUSER`, `$ELUXPASSWORD` and `$ELUXDOMAIN` are used to log on to the authentication server. This allows to use the AD logon data of the eLux® desktop for automatic logon to the configured applications (single sign-on).
	For further information, see User variables.
Application restart	The application is immediately restarted after it has been closed either unexpectedly or by the user.

OPTION	DESCRIPTION
Start automatically after	The application starts automatically after eLux has been started. Optionally, you can delay the auto-start process by defining the required number of seconds.
Desktop icon	Provides an additional desktop shortcut for the application (icon and display name)
	(except for PN Agent)
Free Parameters	Individual parameters for program start

Editing application properties

- Open the context menu of the relevant application and click **Properties**.

The **Application Properties** dialog for the application opens. Depending on the application, different properties can be configured.

> **Note:**

> Properties of the selected application can be displayed in the **Properties** windows of the Scout Console. They cannot be modified there.

> For each administrator, you can control the object rights for individual application types. Object rights for the advanced settings and free parameters can be assigned separately.

Defining free application parameters

Free application parameters are individual parameters which can be used to start an application. You can define free application parameters for all applications, except for SAP-GUI and Emulation.

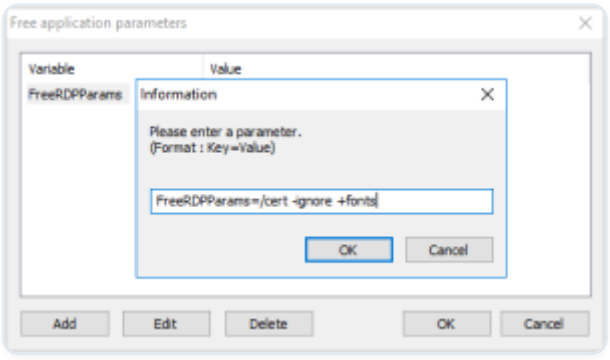
1. Open the **Application properties** of the relevant application.
2. Click **Free Parameters**.
3. Click **Add** and enter one or more parameters in the following format. Example:

   ```
   FreeRDPParams=<Parameter> <Parameter> <Parameter>
   ```

   Separate multiple parameters by spaces.

4. Confirm twice with **OK**.

The defined parameters are saved with the application definition. They will be inserted for the relevant application into the file `\\setup\\sessions.ini`.



Note:

Access to the free parameters can be restricted via the object rights.

For information on which parameters are available, refer to the description of the respective application definition.

The following parameters can be used across applications:

PARAMETER	VALUES	DESCRIPTION
pinned	true	The application is pinned on the system bar.

Parent applications

By default, applications are inherited to subordinate OUs. This allows you to define applications in only few but central places.

For the subordinate OUs, in the tree view, on the **Applications** context menu, the option **Use parent applications > All** is enabled (check mark). With the check mark set, all applications are active that have been defined for higher-level OUs or for the top-level OU. In addition to these applications, you can define more applications valid for this individual OU (and subordinate OUs).

DISABLING INHERITANCE OF APPLICATIONS

1. For an OU that you do not want to receive higher-level applications, open the context menu.
2. Click **Use parent applications > All** to remove the check mark.

The OU cannot use higher-level applications and cannot inherit them to subordinate OUs any longer. Only applications defined within that OU are active.

INHERITING ONLY INDIVIDUAL APPLICATIONS

1. For the OU that you want to receive some of the applications defined for a higher-level OU or at top-level, open the context menu.
2. Make sure that the option **Use parent applications > All** is cleared (no check mark).
3. On the sub-menu **Use parent applications**, under **All**, select the application you want to inherit from above.

The selected application, on the sub-menu, receives a check mark, and its definition is provided on the next device restart for that OU.

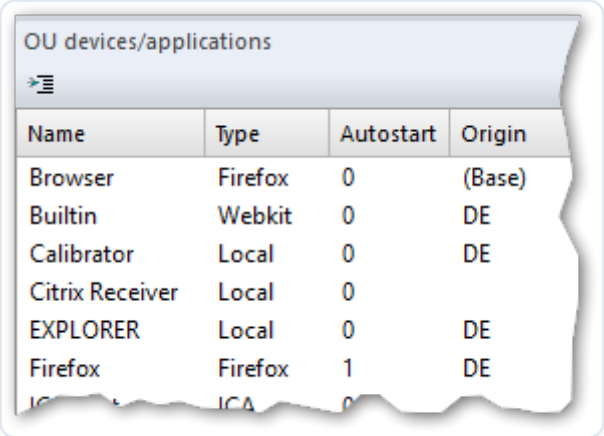
> **Note:**

> Inherited properties of the selected applications will be lost if you move the respective OU to another parent OU. For moved OUs, the system automatically enables **Use parent applications > All**.[1[from Scout 15 2101]]

SHOW DEFINED APPLICATIONS FOR AN OU

1. Click **View > Window > OU devices/applications** to display the relevant window.
2. In the tree view, click the **Applications** icon below an OU.

For the selected OU, all defined applications are listed. The **Origin** column shows the OU from which an application is inherited. Top-level applications show the value Base.


OU devices/applications

Name	Type	Autostart	Origin
Browser	Firefox	0	(Base)
Builtin	Webkit	0	DE
Calibrator	Local	0	DE
Citrix Receiver	Local	0	
EXPLORER	Local	0	DE
Firefox	Firefox	1	DE

The selected OU of the figure above has one own application (no entry in the **Origin** column), four applications from the higher-level OU **DE**, and one top-level application.

> **Note:**
>
> To also apply the default settings of the parent applications, on the **Applications** context menu, select the **Use parent defaults** option. For further information, see **Software defaults**(app_defaults.htm).

Software defaults

Software default settings for all applications of the same type can be defined centrally or on OU-level. Software default settings are available for

- Citrix applications (Citrix Workspace-App) For further information, see Citrix software defaults.
- Browsers For further information, see Browser home directory

We recommend to apply the default settings at top level (root applications) to use inheritance across all OUs. If you still need different software default settings, you need to disable inheritance for the relevant OUs.

DEFINING SOFTWARE DEFAULTS

1. In the tree view, for the relevant OU, open the **Applications** context menu and click **Software defaults**

   > **Note:**

> Citrix Virtual AppsIf inheritance is enabled, you can only open and modify the **Software defaults** of the top-level instance or parent instance. By default, this is the top-level instance.

2. In the list-field, select the relevant software and click **Edit**.
3. Edit the relevant options on the tabs and confirm with **OK**.

The software defaults are applied to all applications of subordinate OUs, provided, on the OU's **Applications** context menu, the **Use parent defaults** option is selected. This is the default.

DISABLING INHERITANCE OF SOFTWARE DEFAULTS

1. In the tree view, for the relevant OU, open the **Applications** context menu.
2. Disable the **Use parent defaults** option.

The parent software defaults are saved for the OU.[1[ab Scout 15 2204]] On this basis, from now on, you may configure the settings individually.

Defining application templates

As an administrator, you can create a template for each application type, for example, a browser template: In the template, you specify values that users will find filled out when they create a browser application. If you create a browser template and specify the browser type **Firefox** and **System proxy** as proxy, as a result users at all levels will see Firefox and system proxy as predefined values when they create a new browser application.

Note that the values can be overwritten. However, if you disable the object rights for the advanced options or free parameters of an application type, these values remain because the relevant users do not have access to them in the application definition.

Application templates are defined at the top level. You can define exactly one template for each application type.

CREATING AN APPLICATION TEMPLATE

REQUIRES
Administrators' base right: **Edit base application**

1. In the tree view, on the top level, right-click the **Applications** icon.
2. On the context menu, click **Define application templates**

   The **Definition of application templates** dialog opens. Each tab contains the application properties of an application type.
3. Switch to the tab that contains the required application type.
4. Edit any options you want to predefine.
5. Confirm with **Apply** and **OK**.

For each new application definition of this application type, the predefined values will be set.

MODIFYING OR RESETTING APPLICATION TEMPLATES

1. In the tree view, on the top level, right-click the **Applications** icon and select **Define application templates**

   The **Definition of application templates** dialog contains for each application type a tab with its application properties. Non-empty fields contain values that are used when new applications of this type are created.
2. For the relevant application type, edit the options you want to change.
3. To reset all values that you have predefined for an application type, click **Reset**.

Uploading applications from a device to Scout

Application definitions of a reference device with an up-to-date eLux® version can be uploaded to the Scout Console and assigned to any OU.

> **Important:**
>
> If you upload applications to an OU, all existing applications in this OU will be deleted.

UPLOADING FROM ANY DEVICE (OUTSIDE OF SCOUT ENTERPRISE MANAGEMENT SUITE®)

1. In the Scout Console, click **File > Application upload**.

The **Application upload** dialog opens.

1. Enter the IP address or name of the device you want to upload application definitions from.
2. Select the **Destination** OU.
3. Click **Start**.

The application definitions of the specified device (or its OU) are uploaded to the specified OU. Already existing applications are deleted.

UPLOADING FROM A DEVICE MANAGED BY SCOUT ENTERPRISE MANAGEMENT SUITE

1. In the Scout Console, select the device you want to upload application definitions from.
2. Click **File > Application upload**.

The **Application upload** dialog opens. The IP address of the selected device is already set in the field **IP-name or IP-address of the device**.

1. Select the **destination** OU to which the application definitions are to be copied.
2. Click **Start**.

The application definitions of the specified device (or its OU) are uploaded to the specified OU. Already existing applications are deleted.

Defining application icons

You can define custom icons for applications to be displayed on the devices\' desktop. For the icon files, the high-resolution formats `.svg` and `.png` are supported.

1. In the tree view, for the top-level **Applications**, from the context menu, choose **Define application icons.**.
2. Click **Add** and select the relevant file from the file system.
3. Confirm with **Open** and **OK**.

The application icon is shown in the dialog. The icon is defined but has not been assigned yet.

Assigning custom application icons

> **Note:**

> Before you assign an application icon other than the default icon, make sure that the icon is already defined. For further information, see Defining application icons.

1. For the relevant application, from the context menu, choose **Properties...**.
2. Select the **Desktop icon** option.
3. Click **...** and select one of the icons.
4. Confirm with **OK** and **Apply**.

The application icon is shown for the selected application on the next device restart.

Defining a service app

> **Note:**
>
> To use this feature, user authentication via Active Directory is required.

> If you use AD, you can allow users to start eLux® in service mode without logging on. To do so, define one or more service apps. The AD logon dialog then provides an additional **Service** button that starts eLux in a protected mode (service mode). In service mode, eLux only offers the defined service apps on the desktop. In the Configuration panel, only the **Information** dialog is shown.

1. To define a service app, open the **Application properties** of the relevant application.
2. Click **Free Parameters** and add the following parameter:

   `ServiceApp=true`
3. Confirm with **OK**.

After the parameter is transferred to the device, the AD logon dialog contains a **Service** button.

Limiting applications to one logon domain

> **Note:**
>
> To use this feature, user authentication via Active Directory is required.

> If users have configured multiple AD domains for log-on, you can limit the display of an application to one of the configured domains. This option is defined via a free parameter in the application definition.

1. Open the **Application properties** of the relevant application.
2. Click **Free Parameters** and add the following parameter:

   `ShowInDomain=<AD logon domain>`

The AD logon domain must match with one of the domains specified in the device configuration under **Security > User authentication > AD directory**.

Example: `int.sampletec-01.com`

3. Confirm with **OK**.

After the parameter is transferred to the device, the relevant application is only shown if users log on to the specified AD domain.

Control Application Visibility with the ShowApp Parameter

Goal

This guide explains how to use the **ShowApp** parameter in eLux to control whether an application is visible to users based on device properties or environmental conditions. Administrators can define conditional expressions so that applications appear only when specific criteria are met.

This capability helps administrators:

- Restrict applications to certain devices or hardware models
- Show applications only for specific OS versions or configurations
- Hide applications dynamically when device conditions change

The feature was introduced in **eLux 7 2601.0**.

Prerequisites

Before configuring **ShowApp**, ensure the following:

- You have **administrative access to Scout**.
- The application is already **configured in Scout**.
- You can modify **application free parameters**.
- Devices are running **eLux 7.2601 or later**.

How ShowApp Works

The **ShowApp** parameter contains a **boolean expression** that evaluates device properties.

An application will only appear on the user's desktop if the expression evaluates to **true**. If the expression evaluates to **false**, the application is hidden.

Each condition follows this format of `Property(GLOB)`:

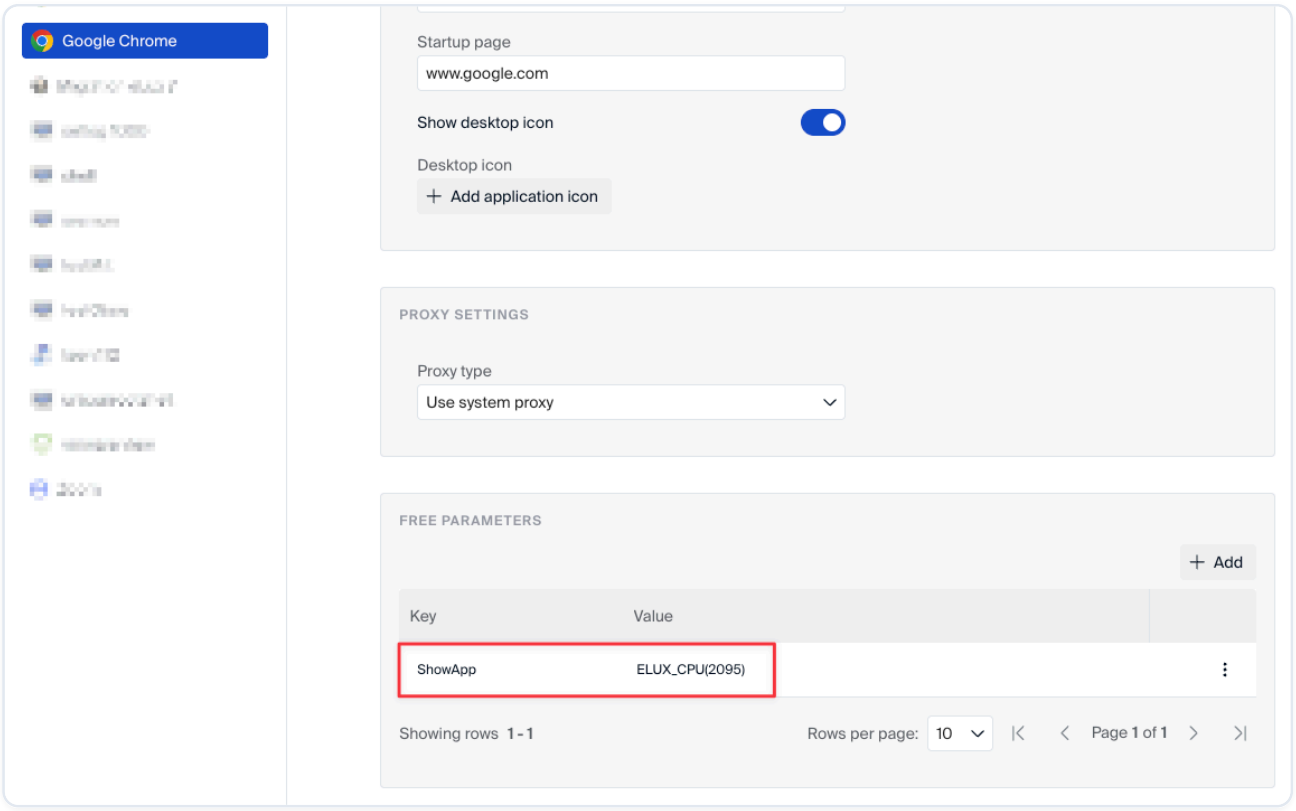
- **PROPERTY** – a device attribute such as hostname, OS version, CPU, or memory.

- **GLOB** – a pattern used to match the value (supports wildcards like `*`).

Multiple conditions can be combined using boolean operators:

- `AND`
- `OR`
- `NOT`
- Parentheses `()` for grouping expressions

Step-by-Step: Configure ShowApp in Scout



1. OPEN THE APPLICATION CONFIGURATION

1. Open **Scout**.
2. Navigate to the desired **device profile**.
3. Select the **application** you want to control.

2. ADD THE SHOWAPP PARAMETER

1. Go to **Free parameters** for the application.
2. Add the parameter: `ShowApp=`
3. Replace `<expression>` with your conditional logic.

Example: `ShowApp=ELUX_HOSTNAME(DellPrecision*)`

This configuration shows the application only on devices whose hostname begins with **DellPrecision**.

3. DEPLOY THE CONFIGURATION

1. Save the profile configuration.
2. Apply the profile to the relevant devices.
3. Restart or refresh the device configuration if required.

The application will appear only when the **ShowApp expression evaluates to true**.

Example Expressions

MATCH A SPECIFIC HOSTNAME

```
ELUX_HOSTNAME(DellPrecision3260)
```

Application appears only on a device with this exact hostname.

MATCH MULTIPLE DEVICES WITH A WILDCARD

```
ELUX_HOSTNAME(DellPrecision*)
```

Application appears on any device whose hostname begins with **DellPrecision**.

COMBINE MULTIPLE CONDITIONS

```
ELUX_OSVERSION(7.2507.0-3) AND ELUX_SERIAL(71T0ZP3) AND ELUX_MEMORY(8192)
```

Application appears only when:

- OS version matches
- Serial number matches
- Device has exactly **8192 MiB RAM**

All conditions must be satisfied because of the **AND** operator.

COMPLEX EXPRESSION EXAMPLE

```
ELUX_CPU(3900) OR ELUX_OSNAME(eLux) AND (ELUX_OSVERSION(7.2507.0-3) OR ELUX_SERIAL(71T0ZP3))
```

This example combines multiple conditions using both **AND** and **OR** operators.

Available Device Properties

Common properties that can be used in **ShowApp** expressions include:

- `ELUX_HOSTNAME`
- `ELUX_OSNAME`
- `ELUX_OSVERSION`
- `ELUX_SERIAL`
- `ELUX_SUPPLIER`
- `ELUX_DEVICETYPE`
- `ELUX_BIOS`
- `ELUX_CPU`
- `ELUX_PRODUCT`
- `ELUX_KERNEL`
- `ELUX_FLASH`
- `ELUX_IDF`
- `ELUX_MAC`
- `ELUX_IP`
- `ELUX_NETMASK`
- `ELUX_BROADCAST`
- `ELUX_NETADDR`
- `ELUX_FLASHSIZE`
- `ELUX_MEMORY`
- `ELUX_GRAPHICS`

These values are collected by the system during **boot and configuration updates**.

Troubleshooting

The evaluation of ShowApp expressions can be inspected in the following log files.

ELUXD LOG `/TMP/ELUXD.LOG`

Search for entries containing: `showapp`. This log records how the expression was parsed and whether it evaluated to **true** or **false**.

DESKTOP LOG `/TMP/UCDESKTOP.LOG`

This log records when applications are hidden from the desktop because the **ShowApp expression evaluated to false**.

Important Behavior

- Device property values are refreshed **during boot and configuration changes**.
- If device conditions change while an application is running and the expression evaluates to **false**, the application may be **terminated automatically**.
- The eLux desktop hides application icons when the application is marked as hidden.

Best Practices

- Start with **simple expressions** and test them before deploying complex logic.
- Use **wildcards (`*`)** to simplify hostname or device matching.
- Verify behavior using **log files** during initial testing.
- Document expressions used in **Scout profiles** to simplify maintenance.

Connecting to a Citrix® farm

Users can connect to sessions running on a Citrix back-end. Once the connection has been made, the user can access published desktops and applications.

Connecting a device to a Citrix back-end is performed by one of the following applications:

- by a StoreFront application to a StoreFront server
- by the Citrix Self-Service user interface to a StoreFront server
- via a browser to a StoreFront server or Web Interface server, see Browser session to access published resources

Requirements

- The eLux® package **Citrix Workspace-App for Linux** must be installed on the devices.
- To connect via HTTPS, for the application types **Storefront™, Self Service and PNAgent**, the relevant root and intermediate certificates must be available on the devices.
    - Root certificates must be transferred to `/setup/cacerts`.
    - Intermediate certificates must be transferred to `/setup/cacerts/intcerts`.

For further information, see Certificates in the **Installation** guide.

- To connect via HTTPS, for the application type **Browser**, the relevant root and intermediate certificates must be available on the devices.
    - Firefox: Root certificates and intermediate certificates must be transferred to

    - Chromium: Root certificates and intermediate certificates must be transferred to

- The eLux taskbar should be enabled on the devices if published applications are provided as **seamless applications**. Seamless applications behave like local applications and users can only restore them from minimized window size by using the taskbar. For further information, see Advanced desktop settings.
- StoreFront application
- StoreFront / Store provisioning file
- StoreFront / Authentication
- Self-Service user interface
- Custom design for Citrix Workspace-App
- Browser session to access published resources
- Citrix software defaults
- Citrix Connection Center
- Logging for Citrix Workspace-App
- Updating Citrix Workspace-App

Storefront™ application

By using the application type **StoreFront**, users can connect to a Citrix StoreFront server. Virtual desktops and published applications are aggregated and provided through stores. The Citrix products mainly used are XenApp and Citrix XenDesktop. StoreFront sites can be accessed via HTTP or HTTPS.

The StoreFront application enables users to access Citrix resources of one or more stores together with other configured applications, such as **RDP** or **Browser** sessions by using only one interface -- the User Interface. For further information, see eLux RP 6 User Interface.

DEFINING A STOREFRONT APPLICATION

> **Note:**
>
> HTTPS connections require the relevant Certificates on the device.

1. Add a new application see Adding applications and select the application type **StoreFront**.
2. Edit the following fields:

OPTION	DESCRIPTION
Name	Name of the application shown in the Scout Console
Use Provisioning File (.cr)	Use a provisioning file. This option excludes the specification of Store URLs (next option).
Stores	Specify the URL of one or more stores.
	- Click **Add** and replace the automatically created default value by your individual value (double-click or F2).
	Example: `https://CtrXd76.sampletec-01.com/Citrix/Store33/discovery`
	This option excludes the use of a Provisioning file (previous option).
Logon	The user is automatically logged on to the store by using the specified credentials (username, password, domain).
Pass-through logon	The user is logged on to the store via single sign-on. The AD user credentials are used.

Note:

If you want to use predefined credentials or pass-through authentication, the eLux® package **Citrix Extensions** and the included feature package **Dialog Extension** must be installed on the devices.

For further information, see StoreFront / Authentication.

Show last user	The user credentials (except for password) of the last logon are displayed in the XenApp® logon dialog. This option has no

	effect if you specify fixed user credentials for automatic logon under **Logon**
Autostart	Specify the names of those StoreFront applications you want to have started automatically. Make sure to spell the names exactly as in StoreFront. Separate multiple application names by semicolon. Example: `MyApp1;MyApp2`. If only one resource is defined for a store, alternatively use the free parameter .
Application restart	See Adding applications.
Start automatically	
Desktop icon	
Free parameters (optional)	Individual parameters for application start. For further information, see Defining free application parameters.

3. To delete an entry from the **Stores** list, select the entry and click **Delete**.
4. To configure further settings, click **Advanced** and edit the following fields:

OPTION	DESCRIPTION
Windows properties	Desktops can be launched in full-screen or window mode.
Timed logoff	To enable automatic logoff from the StoreFront server, select the **Logoff after** option and specify a delay in seconds. Automatic logoff does not affect the launched desktop. Alternatively, automatic logoff can be configured to be performed after the last StoreFront application has been closed.
Application reconnection	Determine the actions to be done on a reconnect to the StoreFront server

OPTION	DESCRIPTION
	Do not reconnect: The connection to the desktop or the published applications is not restored (default).
	Disconnected sessions only: The connection to a disconnected session is restored.
	Active and disconnected sessions: The connection to a disconnected or active session is restored.
Manual logoff	Determine the actions to be carried out upon logoff from the StoreFront server
	Logoff only server: Logoff is performed only from the StoreFront server
	Logoff server and applications: Logoff is performed from the StoreFront server and from the virtual desktop or published applications.
	Logoff server and disconnect session: Logoff is performed from the StoreFront server but the virtual desktop session is only disconnected. This enables the user to reconnect later on.

> **Note:**
>
> Access to the advanced settings can be defined via the object rights.

5. Confirm with **Apply** and **OK**.

After users have logged on to a StoreFront server or Web Interface server, they can show all provided resources by double-clicking the **StoreFront** icon on the eLux desktop.

StoreFront / Store provisioning file

A Citrix store provisioning file can be created by the Citrix back-end and contains all relevant connection information. Using this file allows to switch automatically from Citrix StoreFront™ connection data to Citrix Access Gateway connection data if StoreFront is not reachable (scenario of switching between company office and home office).

To use a store provisioning file for your eLux® devices, note the following:

- The file must be located on the devices in the directory `/setup/ica/` Transfer the .cr file by using the Scout feature . For further information, see Files configured for transfer.

StoreFront / Authentication

If on a device, smart card packages are installed and the Citrix Workspace-App for Linux identifies smart card middleware on the device, smart card logon has precedence by default. In order to still authenticate via username and password for certain devices, define the authentication method via a parameter:

CONTROLLING THE AUTHENTICATION METHOD VIA ELUX®

The logon method can be changed to username and password regardless of the smart card packages installed.

- Define the following entry by using the Scout feature Adding individual file entries:

File	`/setup/sessions.ini`
Section	`ICADefaults`
Entry	`StoreFrontLogOnWithPassword`
Value	`true | false` , (Default: false)

CONFIGURING SMART CARD BEHAVIOR

If you use smart card authentication for StoreFront™, you can configure the behavior of the smart card when it is removed.

> **Note:**

> Using a smart card requires the smart card middleware to be installed on the device. In addition, smart card authentication must be enabled on the Citrix® farm.

- Define the following entry by using the Scout feature Adding individual file entries:

File	`/setup/sessions.ini`
Section	`ICADefaults`
Entry	`SmartcardRemovalAction`
Value	`noaction | forcelogoff` , (Default: noaction)

FURTHER PARAMETERS FOR AUTHENTICATION

To define further parameters for authentication, use the configuration file `/setup/ica/AuthManConfig.xml.template`. This file is transferred to the devices during installation. Using the **Diagnosis** and **Files configured for transfer** features, you can retrieve the file, edit it and transfer it back to the relevant devices. For further information, see Files configured for transfer.

For the function to become active on the device, one restart is necessary to transfer the file, and another restart to activate the new parameters.

Self-Service user interface

The Self-Service user interface (UI) replaces the configuration manager **wfcmgr** and allows access to Citrix services providing published resources.After users are set up with an account, they can subscribe to desktops and applications, and then start them.

DEFINING CITRIX SELF-SERVICE AS LOCAL APPLICATION

> **Note:**

> The eLux® package **Citrix Workspace-App for Linux** and the included feature package **Self-service** must be installed on the devices. This may require modifications of the image definition file on the web server via ELIAS.

1. Add a new application and select the application type **Local**.

2.

OPTION	DESCRIPTION
Name	Name for the application
Local application	Select `Custom`.
Parameter	Enter the following program name to start the application:
(mandatory)	`selfservice`

3. Confirm with **Apply** and **OK**.

> **Note:**
>
> The `selfservice` application cannot be configured individually. To use configuration options, alternatively use the Citrix Self-Service in kiosk mode.

Custom design for Citrix Workspace-App

> **Note:**
>
> The eLux package **Citrix Workspace-App** (version 18.08 or later) must be installed on the devices.

> To customize the layout of your Citrix® session, transfer the relevant layout files to the devices into the Citrix directory structure. The files then are merged with the Citrix layout files.

To transfer the files, use the Scout feature **Files configured for transfer**. For further information, see Files configured for transfer.

As destination specify the provided Citrix directories. Example:

Default	`/setup/ica/site_custom`
If **Shared User Mode** is used	`/setup/ica/site_custom/sum_screen`

The Citrix directory structure must be retained.The original structure can be found under `/opt/Citrix/ICAClient/site_orig`.

Browser session to access published resources

Users can access applications and desktops that have been published through a store on the Citrix StoreFront™ server or through Citrix Web Interface by using a local browser.

DEFINING A BROWSER APPLICATION TO ACCESS PUBLISHED RESOURCES

> **Note:**
>
> To provide the users with a browser application to be used directly on the device, the relevant software package for Firefox or Chromium must be installed on the devices. This may require modifications of the image definition file on the web server via ELIAS.

> HTTPS connections require the relevant SSL certificates on the device.

1. Add a new application see Adding applications and select the application type **Browser**.
2. Edit the following fields:

OPTION	DESCRIPTION
Name	Name for the browser session
Browser type	Firefox or Chromium
Called page	URL of the Web Interface homepage or StoreFront store.
	Examples
	`https://<Servername>/Citrix/StoreWeb`
	`https://<Servername>/Citrix/XenApp`

3. For the remaining parameters, see Defining a browser application.

The local user starts the browser and is forwarded to the defined page. After successful logon to the StoreFront server or Web Interface server, the published applications, desktops and contents available are shown in the browser window.

Citrix® software defaults

For all Citrix applications, in the Scout Console, you can define Citrix Workspace-App software defaults that are applied to all devices of the relevant OU and subordinate OUs if configured.

The following options are available:

- Client drive mapping
- COM port mapping
- Firewall settings
- Citrix keyboard shortcuts
- Window properties
- Connection options
- Bitmap caching

To edit the software defaults, see Software defaults.

> **Note:**

> To define parameters in individual configuration files, use the Advanced file entries feature. All parameters defined by using the **Advanced ini entries** override the software defaults.

> Some of the Citrix default options are described below. For further information, see the Citrix documentation.

GENERAL TAB

OPTION	DESCRIPTION
TW2StopwatchMinimum	Scrolling speed for remote applications (such as Adobe Acrobat Reader, Microsoft Excel)
	The higher the value, the slower the speed when scrolling

OPTION	DESCRIPTION
	Note for Excel: A low value increases scrolling speed but delays as soon as a selection is drawn down out of the visible screen area.
	Default = 25
Client name template	Definition of the client name in the Citrix session
	Note: You can use the Program Neighborhood variables [$ICANAME] and [$ICADOMAIN] to set a unique client session name. This is required for Citrix Roaming and some XenApp programs. For further information, see PNAgent application.

DRIVE MAPPING TAB

OPTION	DESCRIPTION
A-Z	The letters A to Z represent the logical drive names on the terminal server. In the field on the right, you can assign a local resource to a drive letter that is to be shown in the Citrix session.
	Enter the mount point relating to the local access path of the resource. The mount points are provided by eLux: [/media/usbdisk] or [/media/cdrom]
Attributes E / R / W	Type of access right
	• E = enable
	• R = read
	• W = write

OPTION	DESCRIPTION
Enable Drive Mapping	Must be selected to enable the defined drive mappings
Enable Dynamic Mapping	Available mass storage devices are assigned to the next free drive letter.

For further information, see Mount points.

COM PORTS TAB

To connect via COM port, the device name of the client device's COM port is required.

The COM port device name always begins with the string . Device names are case-sensitive.

Examples:

PORT DEVICE NAME	COM PORT
[/dev/ttyS0]	COM1
[/dev/ttyS1]	COM2

The availability of COM ports depends on the hardware platform.

> **Note:**

> The devices' ports must be mapped on the Citrix resource (such as desktops) accordingly. To do so, use a `net use` command. Example: `net use com1: ClientCOM2: /persistent:yes`.

Citrix Connection Center

By means of the Citrix Connection Center, users can see all current server connections and can log off, disconnect or close them without operating the application.In addition, the connection transport statistics can be viewed which might be helpful for slowing connections.

The Connection Center is provided as local desktop application.

DEFINING THE CITRIX CONNECTION CENTER

> **Note:**

> The **Citrix Workspace-App** and the included feature package **Utilities and tools** must be installed on the devices. This may require modifications of the image definition file on the web server via ELIAS.

1. Add a new application see Adding applications and select the application type **Custom**.
2.

OPTION	DESCRIPTION
Name	Citrix Connection Center
Local application	Select `Custom`
Parameter	`/opt/Citrix/ICAClient/util/conncenter`
Desktop Icon	Activate desktop icon

3. Confirm with **Apply** and **OK**.

Logging for Citrix Workspace-App

For the Citrix Workspace-App, you can enable and configure logging via a configuration parameter.

CONFIGURING THE LOG LEVEL

1. In the Scout Console, for the relevant devices, open **Advanced device configuration > Advanced file entries**.
2. Define the following entry

FILE	
Section	
Entry	
Value	0|3|7
	0: Logging disabled (default)
	3: Only errors are logged

FILE	
	7: Logs for all levels are generated

For further information, see Advanced file entries.

As soon as the new configuration is active, logs are written to .

ENABLING DETAILED LOGGING FOR MICROSOFT TEAMS

> **Note:**
>
> The eLux® package **Citrix Workspace-App for Linux** and the included feature package **Microsoft Teams Optimization** must be installed on the devices.

1. Create a configuration file named `config.json` and enter the following:

```
[Copy]



{

  "WebrtcLogLevel" : 0,

  "WebrpcLogLevel" : 0

}
```

2. Transfer the `config.json` file to the devices to `/setup/ica/hdx_rtc_engine`

   For further information, see Files configured for transfer.

When Microsoft Teams (in VDI) is used on the devices, the relevant log files are written.

- To access the log files, use the diagnostics function. To do so, create a diagnostic template that contains the following entries:

  /tmp/hdxrtcengine// /tmp/webrpc//

For further information, see Device diagnostics.

After retrieving the diagnostic files, you can find the log files in the `hdxrtcengine` and `webrpc` directories.

Updating Citrix Workspace-App

- To install later versions of the Citrix Workspace-App on the devices, perform a firmware update.

> **Note:**
>
> The existing `/setup/ica/AuthManConfig.xml` template will be overwritten with the template of the new version. Individual entries must then be set again.

BEHAVIOR OF DEFAULT VALUES

-from Scout 15 2204

When you update to a later version of the Citrix Workspace-App, the vendor's default values are applied. This overwrites individually set values in favor of a consistent situation between updated devices and freshly installed devices.

The vendor's default values are set for the following scenarios:

- Firmware update to newer CWA package (with and without formatting the system partition)
- Firmware update to newer eLux® version including new CWA version (with and without formatting the system partition)
- Factory reset
- PXE or USB installation

RDP

The **RDP** application type uses the Microsoft Remote Desktop Protocol (RDP) to connect to a Microsoft terminal server. The provided RDP client is eLuxRDP that is based on the free software implementation **FreeRDP**.

There are two options for configuration:

- Windows Desktop: The user accesses the desktop of a terminal server by using a remote desktop session. The user can use any application available on the desktop.
- Individual / seamless application: The user can only access one particular application of the terminal server.

- Defining an RDP Windows desktop session
- Defining an RDP application
- Defining a load-balanced RDP session
- Advanced application settings / RDP and VMware
- Configuring RemoteFX

Defining an RDP Windows desktop session

1. Add a new application see Adding applications and select the application type **RDP**.
2. Edit the following fields:

OPTION	DESCRIPTION
Name	Name for the RDP application
Server	IP address or name of the server
Application	Leave the field empty.
Working directory	Leave the field empty.
Logon	The user is automatically logged on to the server by using the specified credentials (username, password, domain).
	The user is logged on via single sign-on. The AD user credentials are used.
Free parameters	Allows to define any parameters supported by **eluxRDP** in the format: `FreeRDPParams=<Parameter> <Parameter> <Parameter>`. Separate multiple parameters by spaces. Examples: `FreeRDPParams=/microphone:sys:pulse +fonts /cert-ignore`. To view the allowed parameters, enter the `eluxrdp` command in a shell. For further information, see Defining free application parameters.

3. Confirm with **Apply** and **OK**.

> **Note:**
>
> Defining a server-independent application as a local hidden application named `RDP_TEMPLATE` allows you to configure a connection template without a back-end. The user starts `rdpconnect` from the shell and subsequently specifies the server to be connected to. This feature requires the eLux® software package **RDPConnect**.

Defining an RDP

application

To configure an individual application via RDP, the Windows desktop definition requires additional data about the relevant application.

1. Add a new application see Adding applications and select the application type **RDP**.
2. Edit the following fields:

OPTION	DESCRIPTION
Name	Name for the RDP application
Server	IP address or name of the server
Application	Name of the Windows application including path name
	System variables are allowed.
	Examples: `c:\Program Files\Microsoft Office\Office\EXCEL.EXE`
	`%SystemRoot%\system32\notepad.exe`
Working directory (optional)	Working directory of the Windows application
Logon	The user is automatically logged on to the server by using the specified credentials (username, password, domain).
Pass-through logon	The user is logged on via single sign-on. The AD user credentials are used.

OPTION	DESCRIPTION
Free parameters	Allows to define any parameters supported by **eluxRDP** in the format:
	`FreeRDPParams=<Parameter> <Parameter> <Parameter>`

3. Confirm with **Apply** and **OK**.

For the users, the application runs full-screen in the session window.

Defining a load-balanced RDP session

If you are running a remote desktop server farm with load balancing, have your `eLux` RDP sessions load-balanced via the following configuration.

1. Define an RDP session/application or open the **Properties** dialog of an existing one. Edit the following options:

OPTION	DESCRIPTION
Server	Name of your RD connection broker
	Example: `RDPLoadMaster.int.sampletec-01.com`
Free parameters	Parameter for the load balancer server collection
	`FreeRDPParams=/load-balance-info:tsv://MS Terminal Services Plugin.1.<[Load balancer collection]>'`
	Example: `FreeRDPParams=/load-balance-info:tsv://MS Terminal Services Plugin.1.RDS-01>'`

2. Confirm with **Apply** and **OK**.

Advanced application settings for RDP and VMware

The settings described below apply to the following applications:

- RDP applications
- VMware applications

If you select a protocol other than RDP, some options are not available.

ACCESSING ADVANCED APPLICATION SETTINGS

> **Note:**
>
> Access to the advanced settings can be restricted via the object rights.

- Scout: In the **Application properties** dialog of an RDP or VMware application, click the **Advanced** button.
- eLux® RP 6: In the **Application properties** dialog of an RDP or VMware application, under **Properties**, expand the relevant section.

VIEW TAB

OPTION	DESCRIPTION
Window size	Full-screen or a specific resolution
Full-screen on monitor	If you have selected the window size `Full-screen`, select if you want to display on one specific or all monitors. Up to eight monitors are supported.
Colors	Color depth for the session (8-32 Bit)

> **Note:**
>
> If you use multiple monitors but wish to display content on only one of them, under **Device configuration > Desktop > Advanced > Windowmanager**, the **Maximize/fullscreen to single monitor** option must be selected.

LOCAL RESOURCES TAB

for terminal servers supporting RDP protocol version 5.2 or later -- The settings take effect only if, on the **Advanced** tab, the value of the **Protocol** field is not set to `RDP V4`.

OPTION	DESCRIPTION
Drive mapping	Select drive, mount point and drive letter that you want to show in the RDP/VMware session. The mount points correspond to the local access paths of the resources and are provided by eLux. For USB devices, define the assignment with `/media`. Internally, the mount point for USB devices is set to /media/usbdisk. If multiple USB devices are plugged in, they use the same mapping. Each USB device is displayed in its dedicated folder. A digit is appended to the folder name and incremented `(usbdisk0, usbdisk1)`. For further information, see Mount points.
Connect printer	Up to four printer definitions can be created automatically for a session. The printers must be configured on the Defining a locally connected printer. The first four profiles can be used with drivers. To define a default printer, choose **Set as default** in the eLux printer configuration.
Sound	**Play local** reproduces the sound locally on the device. **Play remote** causes playback on the remote server.
Connect ports	Makes the defined port connections accessible in the session.
Enable smartcard	Smart cards based on a certificate can be used for logon.

ADVANCED TAB

OPTION	DESCRIPTION
Protocol (only RDP)	Enables you to set the RDP protocol to version 4 or 5. Normally, the protocol is recognized automatically.

OPTION	DESCRIPTION
Keyboard language	Defines the keyboard layout within a session. The default is **Auto** which corresponds to the keyboard setting of the eLux device configuration.

Important:

If you define a specific language, it must be identical to the keyboard language defined in the eLux device configuration, in the **Keyboard** dialog.

OPTION	DESCRIPTION
Deactivate Window-Manager Decorations	The frames of the eLux windows are hidden.
Deactivate encrypting	The server does not accept encrypted sessions. You can use this option to increase performance. By default, the option is disabled.
Deactivate mouse move events	Mouse position data are not transferred to the server constantly, but with every mouse click. This increases system performance and is especially helpful for connections with small bandwidth. By default, the option is disabled.
Show connection bar on full screen	Shows connection list in full-screen mode.
Bandwidth	Choose between `standard, modem, broadband or LAN`.

Configuring RemoteFX

Microsoft RemoteFX™ offers comprehensive functionality for Virtual Desktop Infrastructure (VDI) by providing a virtual 3D adapter, intelligent codecs and the ability to redirect USB devices to virtual machines.

Note:

RemoteFX only works if the server supports RemoteFX and is configured in the right way. The only parameter to be configured on the device is bandwidth.

1. For your **RDP** application, open the **Application properties** dialog and click **Advanced**.
2. On the **Advanced** tab, in the **Bandwidth** field, select `LAN`.
3. Confirm with **Apply** and **OK**.

Virtual Desktop

Note:

For eLux® RP 6, instead of **Virtual Desktop**, the application type **VMware Horizon** is available.

As Virtual Desktop VMware Horizon is supported.

- Defining a virtual desktop
- VMware Horizon

Defining a virtual desktop

1. Add a new application see Adding applications and select the application type **Virtual Desktop**.
2. Edit the following fields:

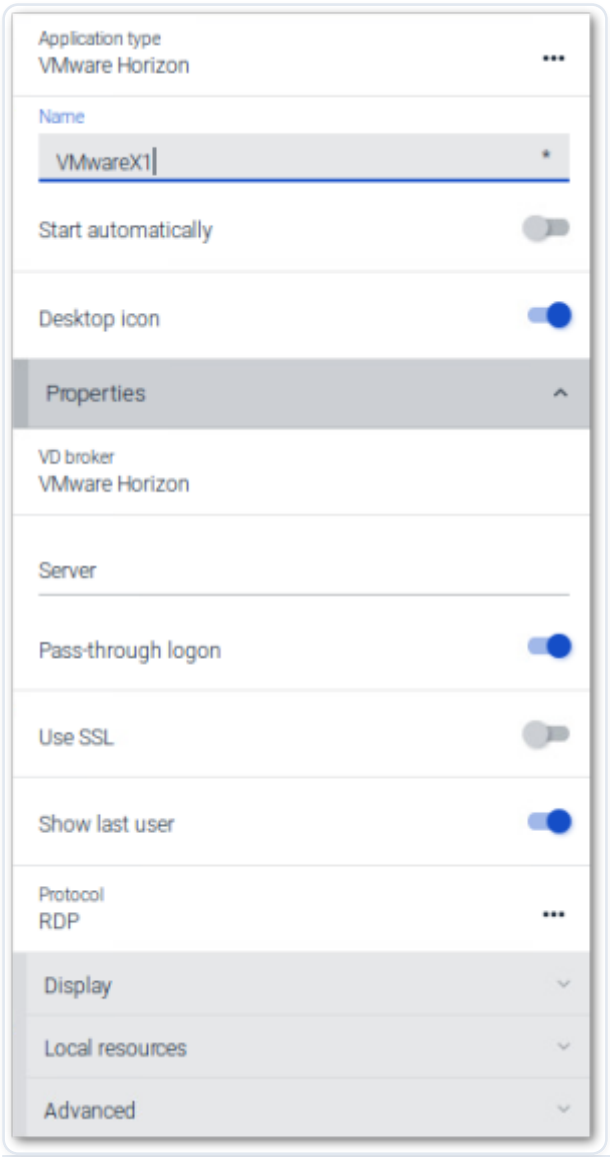
OPTION	DESCRIPTION
Name	Name for the application
VD Broker	Select a virtual desktop application from the list.
Server	Enter the server IP address (or name)
Logon, Pass-through logon	See Adding applications

OPTION	DESCRIPTION
Protocol, (VMware Horizon only)	Choose between the following values: `RDP, PCoIP, VMware Blast`

3. To configure further settings for XenDesktop® or VMware Horizon, click **Advanced**. For further information, depending on the broker selected, see
   - Advanced application settings / RDP and VMware or
   - Advanced XenDesktop settings in PNAgent application
4. Confirm with **Apply** and **OK**.

VMware Horizon

This application type is only available on eLux® RP 6 devices. In the Scout Console, select the Virtual Desktop application type and as VD-Broker VMware View.



OPTION	DESCRIPTION
Name	Name for the application
Auto-start	The application starts automatically after eLux has been started.
Desktop icon	Provides a desktop shortcut on your personal desktop
VD broker	VMware Horizon
Server	IP address or name of the server
Pass-through logon	The user is logged on via single sign-on. The AD user credentials are used.

OPTION	DESCRIPTION
Username, Password, Domain	The user is automatically logged on to the server by using the specified credentials.
Use SSL	Forces the connection via HTTPS
	Note that HTTPS connections require the relevant SSL certificates on the device.
Show last user	The user credentials (except for password) of the last logon are displayed in the logon dialog
Protocol	Choose between the following protocols
	`RDP`
	`PCoIP`
	`VMware Blast`

For information on **Display, Local resources and Advanced** settings, see Advanced application settings.

You may configure the VMware Horizon client by using the application definition in the or locally on the client. To set additional parameters that are not included in the interface, use a configuration file:

- With the help of VMware documentation,[1[Installation guide for VMware Horizon Client for Linux]] create the file `view-userpreferences`. Transfer the file via the Scout feature Files configured for transfer to `/setup/elux/.vmware/view-userpreferences`.

> **Note:**

> The configuration on the Scout or eLux interface has precedence over the configuration file and will overwrite values of the configuration file.

Browser

Supported browsers are Mozilla Firefox and Google Chromium.

In addition, the Builtin Browser is available as a slimmed-down browser. The Builtin Browser is based on the WebKit2 engine which is part of the **Desktop environment** package. By default, the Builtin Browser is run without address and navigation bar. These and some more features can be configured for the kiosk mode.

> **Note:**

> If you use Chromium, we recommend that you equip your devices with 2 GB of RAM.

> For eLux® RP 6, the Java browser plugin is no longer supported.

- Defining a browser application
- Preferences Chromium
- Preferences Firefox
- Preferences Builtin Browser
- Browser home directory
- Kiosk mode for Firefox
- Chrome Enterprise Premium (CEP)

Defining a browser application

1. Add a new application adding a new application and select the application type **Browser**.
2. Edit the following fields:

OPTION	DESCRIPTION
Name	Name of the browser shown in the Scout Console
Browser type	Select `Firefox, Chromium or Builtin Browser`.
Start page	Web page (URL) that opens when you click **Home**
Called page	Web page (URL) that opens after starting the browser

OPTION	DESCRIPTION
Proxy type	- `No proxy`: No proxy server is used
	- `Manual (Proxy:Port)`: Specify a proxy server and port number
	- `Auto (URL)`: Use a proxy configuration file
	- `Use system proxy` (default): 'System-wide' proxy setting defined in the device configuration under **Network > Advanced** per network profile
	Note that the setting behind `System proxy` can also be No proxy.
	For further information, see Proxy configuration.

Note:

For the Builtin Browser, the setting must be left on Use `system proxy`.

NAME	INFO
Application restart	See Adding applications
Start automatically	
Desktop icon	
Free parameters (optional)	Individual parameters for application start
	see Defining free application parameters

3. To enable the **Kiosk** mode for Firefox, see Kiosk mode for Firefox.
4. Confirm with **Apply** and **OK**.

Note:

By default, all browser files (cache, history, bookmarks, etc.) are saved temporarily to the flash memory but are deleted with each restart. We recommend that you configure the browser home directory on a network drive. For further information, see Browser home directory.

Further browser-specific preferences can be set through policies (Chromium) or configuration file entries (Firefox.). For further information, see the Scout guide:

Preferences Chromium

Preferences Firefox

Preferences Builtin Browser

DEPLOYING SSL CERTIFICATES FOR THE BROWSER

- Use the Scout feature **Files configured for transfer** to transfer certificate files to the required target directory on the device:

BROWSER	FOLDER
Mozilla Firefox	`/setup/cacerts/firefox` or `/setup/cacerts/browser`, The certificates can be located in either directory.
Google Chromium	`/setup/cacerts/browser`

For further information, see Files configured for transfer.

Note that a second restart of the device is required to assign the certificates that have been transferred during the first boot to the certificate store of the browser.

Preferences Chromium

By using policies, you can set mandatory (managed) and recommended preferences for the Chromium browser. Mandatory preferences define fixed values that cannot be changed by the user. Recommended preferences define default values that can be changed by the user. For further information, see https://www.chromium.org/administrators/.

- Use the Scout feature **Files configured for transfer** to transfer policy files (.json) to the required target directory on the device:

NAME	FOLDER
Fixed values	`/setup/chromium/managed`
Default values	`/setup/chromium/recommended`

For further information, see Files configured for transfer.

CONFIGURING CHROMIUM POLICIES

If you wnat to define policies fro Cromium and apply them, pleae follow the following steps:

- Create a `JSON` file containing the desired Chromium policies. (Example: `policies.json`)
- Use Unicon OS Management to deploy this file to you UniconOS endpoints. To do this, transfer the policy file to the destination path `/setup/chromium/managed/` using the files section in the advanced device configuration.
- Reboot the enpoint, so that it receives the new configuaion and the policies file
- Open the Chromiom browser with the new policies.

You can verify the active poicies in Chromium by navigating to `chrome://policy/`.

The full list of available Chromium poilicies are documented here: List of Chrome Enterprise policies

Preferences Firefox

For version 60 ESR and later versions, Firefox supports enterprise policies that are deployed via `.json` files and are cross-platform compatible. (This method will not work if Firefox is already being managed by using Windows group policies.) Firefox is installed with enterprise policies enabled that block access to `about:config` and other configuration options by default.

SETTING PREFERENCES WITH .JSON FILES (POLICIES)

from Firefox 60 ESR and later versions

You can use all options that are listed in the README on the Mozilla GitHub repository. (Note that the current Firefox version may differ from that of the eLux version you are using.) One or more options are transferred in a `.json` file to the device by using the Scout feature **Files configured for transfer**.

> **Note:**

> By default, access to the Firefox configuration is blocked.

1. Create a `.json` file (any file name) and insert one or more options separated by commas. Use the syntax shown in the example. (No "policies" brackets!)

   Example:

   ```
   {

     "BlockAboutConfig": false,

     "DisableBuiltinPDFViewer": true

   }
   ```

2. In the Scout Console, for the relevant devices, open **Advanced device configuration > Files**.

   Define your `.json` file as source file. For the destination folder, use `/setup/firefox/policies/`. Example: `/setup/firefox/policies/custom_A.json`

   For further information, see Files configured for transfer.

On the next device restart, the files are transferred and evaluated.

> **Note:**

> You can deploy multiple `.json` files to the device to `/setup/firefox/policies/`. The files are merged in alphabetical order: For identical options, values from files with descending names have precedence (B overrides A).

Preferences Builtin Browser

DISPLAY OPTIONS

By default, the Builtin browser opens with only the title bar, but no navigation bar.

- To display the navigation bar with additional functions, open the properties of your browser application and click **Advanced**. (from 15 2101 and eLux® RP 6 2103)

With which functions the navigation bar is displayed, you also define in the **Advanced browser settings**:

1	Back
2	Forward
3	**Home** button to open home page
4	Refresh
5	Address bar
6	Print

The title bar is always shown.

USER AGENT-STRING

With each page request, the Builtin browser transmits a 'User Agent' string, which is predefined by the WebKit2 engine. It contains information like Mozilla compatibility and operating system.

To transmit individual information, define an Advanced file entry:

1. In the Scout Console, for the relevant devices, open **Advanced device configuration > Advanced file entries**.

File	`/setup/sessions.ini`
Section	`[BuiltinBrowserDefaults]`
Entry	`UserAgentString`
Value	`<name/version> <individual string>`
Example	`Mozilla/5.0 (X11; Linux x86_64)`

   For further information, see Advanced file entries.

> **Note:**
>
> Name and version must be included in the User Agent string.

DNS CACHE

from eLux RP 6 2104

The DNS cache of the Builtin browser is set to 120 seconds by default. To configure the timeout value individually, define an environment variable in the Advanced device configuration:

1. In the Scout Console, for the relevant devices, open **Advanced device configuration > Environment**.
2. To define a new variable, click **New**.
3. Enter the variable using the format: `WEBKIT_DNS_CACHE_EXPIRE_TIMEOUT=<number of seconds>` and confirm with **OK**.

   The new variable is shown in the list.

For further information, see Environment variables.

CLEARING THE BROWSER CACHE

- To delete all browser data including cookies, press `CTRL+SHIFT+DELETE`

Browser home directory

By default, the browser settings are temporarily saved to the flash memory. However, they are deleted with each device restart.

Defining a browser home directory on the network, however, allows you to let users save and make available their browser settings such as bookmarks persistently. To do so, use a network share that you have configured for access:

DEFINING BROWSER HOME DIRECTORY

REQUIRES
Configured Windows network share (Defined drive).
Example: `/smb/share`
For further information, see Defining a network drive.

1. In the tree view, for the relevant level, open the context menu and click **Software defaults**

   For further information, see Software defaults.
2. On the list-field, select the relevant browser and click **Edit**.
3. In the **Browser home directory** field, enter the name of one of the defined drives in **Device configuration > Drives**. The name must correspond to the name on the list. Example: `/smb/share`
4. Confirm with **OK**.

The browser settings are saved to the specified Windows directory.

Kiosk mode for Firefox

for Firefox from ESR version 71.0

> **Note:**

> For current eLux RP 6 versions, you can use the Builtin Browser in kiosk mode. For further information, see Browser in kiosk mode (current eLux versions).

> The kiosk mode starts the browser in full-screen mode and with limited user rights. The user cannot open other windows and cannot exit the browser.

By default, the browser window is displayed without address bar and navigation buttons. So users are forced to stay on the predefined web page and cannot exit.

Kiosk mode is suitable if the users are supposed to see only one website and not use further applications on the device. For correct use of the kiosk mode, we recommend that you disable related functions such as restarting the device and opening the Config panel. For further information, see Device configuration > Security.

CONFIGURING KIOSK MODE

> **Note:**

> Firefox supports kiosk mode again starting with version 71.0, but without configuration options. With Scout 15 2110, the Firefox application definition is adapted and offers only the option **Enable kiosk mode**.

1. In the application properties of your browser application, click .
2. On the **Kiosk mode** tab, edit the following fields:

OPTION	DESCRIPTION
Enable kiosk mode	Activates the kiosk mode
Display navigation bar (up to Scout 15 2107)	Allows using browser tabs despite kiosk mode. Users can view multiple web pages of the defined web site concurrently
Add print button (up to Scout 15 2107)	Allows using browser tabs and provides a **Print** feature despite kiosk mode
Add address bar (up to Scout 15 2107)	Allows using browser tabs and provides the address bar including navigation buttons despite kiosk mode

3. Confirm with **Apply** and **OK**.

On the next restart, the Firefox browser opens in kiosk mode.

Chrome Enterprise Premium (CEP)

Chrome Enterprise Premium (CEP) lets your Google Admin console manage Google Chrome extensions on the device. By default, eLux® blocks all Chrome extensions. Enabling CEP removes this block, so extensions such as Secure Enterprise Browser and Endpoint Verification can be force-installed centrally from your Google Admin console.

CEP is enabled per image, not per application or device configuration. For the enabling steps, see Enabling Chrome Enterprise Premium (CEP) in the ELIAS documentation.

> **Note:**

> CEP requires Google Chrome version 149.0.7827.196.7.2607.0-3 or later. It is switched off by default.

> **Important:**

> CEP and SSO Support for Google Chrome cannot be enabled at the same time. CEP already provides single sign-on for Citrix Secure Access™. If CEP is enabled, SSO Support for Google Chrome must be disabled in the image definition.

USING CEP AFTER ENABLING IT

1. In your Google Admin console, force-install the extensions you want on the affected users or devices, for example Secure Enterprise Browser and Endpoint Verification.
2. Update the device with the image that has CEP enabled, and let it restart.
3. Open Google Chrome on the device.

   The extensions force-installed from your Google Admin console install automatically.

To confirm CEP is active, open `chrome://policy` in Google Chrome and check that the extension force-install policies from your Google Admin console are listed.

local and user-defined applications

Defining local commands is particularly important as they enable the definition of applications which can be launched within a shell. This feature assumes knowledge about the commands that average users may not have.

> **Note:**

> Make sure that the user is authorized to start the application. All commands are executed by the UNIX user **eLux®** (UID = 65534).

> Some of the local applications are predefined. If an application is missing, you can define your own application or command via the Custom option of the **Local Application** list-field.

Error messages will not be shown. If your command does not include an X client application, no messages are shown during execution. For this reason, we recommend first running the command within an XTerm session for testing purposes.

- Defining predefined local applications
- Defining custom applications
- Defining Zoom for Linux
- Defining Ekiga SIP Softphone

Defining predefined local applications

1. Add a new application see Adding applications and select the application type **Local**.
2. Edit the following fields:

OPTION	DESCRIPTION
Name	Name of the application shown in the Scout Console
Local application	In the list-field, select a predefined application.
Parameter (optional)	Command-line parameters for application start
Application restart	See Adding applications
Start automatically	
Desktop icon	
Free parameters (optional)	Individual parameters for application start
	see Defining free application parameters.

3. Confirm with **Apply** and **OK**.

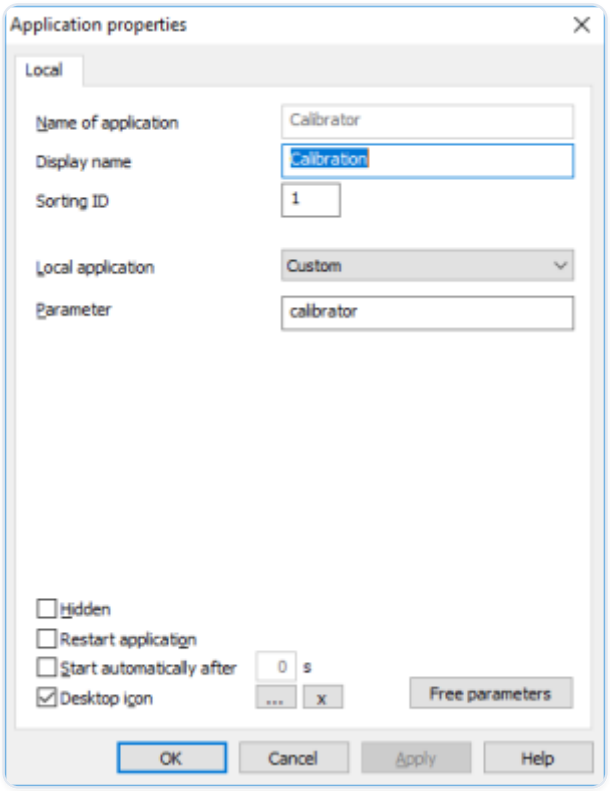
Defining custom applications

1. Add a new application see Adding applications and select the application type **Local**.
2. Edit the following fields:

OPTION	DESCRIPTION
Name	Name of the application shown in the
Local application	Select `Custom`.
Parameter	Enter the program name required to start the application. (mandatory) If required, add start parameters. Example: `calibrator` calls the **Calibrator** tool, `squid` calls the **Squid**

OPTION	DESCRIPTION
	application, `squid /tmp/mycache` calls **Squid** using the specified cache directory
Hidden	The application is not shown on the desktop. This is meant for automatically started applications. Select **Start automatically or Application restart** instead.
Application restart	See Adding applications.
Start automatically	
Desktop icon	
Free parameters	Individual parameters for application start, see (/en-us/unicon-elux-scout/2611/admin_guides/scout/app_definition/app_general/app_params.html)

3. Confirm with **Apply** and **OK**.



The figure shows the application definition for the **Calibrator** tool . After the next device restart, the **Calibration** application is provided on the device's desktop and can be started (provided that the **Calibrator** tool is included in the image).

Defining Zoom for Linux

The native Zoom client for Linux is a Video Conferencing and Web Conferencing service and offers high-quality video, audio, and screen-sharing experience.

The video and audio devices are configured via the application interface.

1. Add a new application see Adding applications and select the application type **Local**.
2. Edit the following fields:

OPTION	DESCRIPTION
Name	Name for the application
Application	Custom
Parameter	zoom

3. Confirm with **Apply** and **OK**.

Defining Ekiga SIP Softphone

Ekiga is a free software application for audio and video telephony (VoIP) that supports the SIP protocol. Configuration is based on a SIP account.

1. Add a new application see Adding applications and select the application type **Local**.
2. Edit the following fields:

OPTION	DESCRIPTION
Name	Name for the application
Application	`Custom`
Parameter	`ekiga`

3. Click **Free parameters** and then **Add** to define the following free parameters:

VARIABLE	VALUE
account	`<Name of the SIP account>`
server	`<server URL>`
user	`<SIP username>`
password	`<password>`
outbound_proxy	`<proxy URL>`

Example: `password=424242` For further information, see Defining free application parameters.

4. In the **Free application parameters** dialog, right-click the variable name `password` and click **Encrypt**.
5. Confirm with **Apply** and **OK**.

Emulation

Applications of the Emulation type can only be defined in the Scout Console but not locally on eLux® RP 6 devices.

Currently, the **5250 terminal emulation** is still supported.

Applications in kiosk mode

Some applications can be configured so that users can only operate them in kiosk mode.

In kiosk mode, the application opens as a full-screen window. Users cannot open any additional windows and cannot close the application. The connection settings and the resources that users are allowed to access at the back-end, such as Citrix® stores, are predefined in a configuration file. As a consequence, defining applications for the user becomes obsolete.

After system start, the device starts the application defined by the configuration files in kiosk mode, ensuring that the user is directly in the intended work environment and cannot break out.

Version information for kiosk mode support

Configuration via .ini file Configuration via console (GUI)

Citrix Self-Service User interface with extensions	from eLux RP 6.2	from Scout 15 2107 and eLux RP 6 xxxx (not yet specified)
Builtin Browser	From eLux RP 6.5[\^1]	from Scout 15 2107 and eLux RP 6 xxxx (not yet specified)
Configuration file	`kioskmode.ini`	`kioskconfig.ini`

: For earlier versions, use Firefox in kiosk mode.

- Setting up kiosk mode via console
- Configuring browser for kiosk mode
- Configuring Citrix Self-Service for kiosk mode
- Citrix Self-Service in kiosk mode (current eLux versions)
- Browser in kiosk mode (current eLux versions)

Citrix Self-Service in kiosk mode (current eLux® versions)

The Citrix Self-Service user interface with extensions is configured as kiosk mode. All relevant parameters are defined in the `kioskmode.ini` configuration file. Advanced functionality and MultiStore option are also configurable.

Defining Citrix Self-Service UI in kiosk mode

> - The eLux package **Citrix Workspace-App for Linux** must be installed on the devices.
> - The eLux package **Citrix Extensions 2.x** or later and the included feature package **Self-service wrapper** must be installed on the devices.
> - For modifications on the Citrix dialog design, further feature packages must be installed on the devices: **Dialog Extension** and **Self-service dialog themes**.
> - This may require modifications of the image definition file on the web server via `ELIAS`.

1. For the relevant devices, open and define the following entry:

File	`/setup/terminal.ini`

Section	Layout
Entry	`KioskMode`
Value	1

For further information, see Advanced file entries.

This parameter enables the kiosk mode for the Citrix Self-Service user interface with extensions.

2. Create a text file named `kioskmode.ini` and add the section header `[Parameters]`. Enter the relevant parameters:[1[Default values are displayed in bold]]

PARAMETER	DESCRIPTION
ReconnectOnLogon=true|`false`	Determines whether the Self-Service UI tries to reconnect to all sessions for a given store, immediately after logon to that store (Citrix Self-Service option)
ReconnectOnLaunchOrRefresh=true|`false`	Determines whether the Self-Service UI tries to reconnect to all sessions when an application is launched or the store is refreshed (Citrix Self-Service option)
SharedUserMode=true|`false`	If the Shared User Mode is enabled, the Self-Service UI uses one technical user account for multiple users. The user data are removed from the device when users log off or close the UI. (Citrix Self-Service option) We recommend that you do not combine and
SelfSelection=true|`false`	Is used to disable the search box and the Self-Selection panel (legacy UI). Disabling these UI elements prevents users from subscribing to extra applications.
LogoffMode=`0`|1|2|3	0 = No automatic logoff 1 = Logoff timer is started with logon 2 = Logoff timer is started when the last Citrix app/desktop is closed 3 =

PARAMETER	DESCRIPTION
	Logoff timer is started when the first Citrix app/desktop is opened
LogoffDelay=`<seconds>`	Delay in seconds after the logoff timer is started
LogoffInfoTimeout=`<seconds>`	Shows a message for n seconds during logoff
ShowLastUser=true|`false`	Shows last logged-on username in **User** field
PreLogin=true|`false`	Determines, whether on logon a dropdown list with pre-configured stores is shown (MultiStore). We recommend that you do not combine `SharedUserMode=false` and
PreLoginTitle	Dialog title for the stores list (MultiStore)
PassThroughMode=`0` |1	0 No pass-through logon data 1 Active Directory UserPassword
Domain=`<Domain display name,domain>`	Each entry contains a domain.

3. To define access to the stores, in the `kioskmode.ini` file, add one or more sections named `[Store<1-N>]` or `[Environment_Store<1-N>]`.

   If you define stores by using `[Environment_Store<N>]`, users can switch between the stores by clicking the **Store** button on the taskbar. The Stores are shown in groups (Environment) as defined.

PARAMETER	DESCRIPTION
Url=`<Store URL>`	URL of the Citrix store
FriendlyName=`<Store display name>`	Display name for the Citrix store
Default=true|`false`	Determines whether this store is displayed as default store

PARAMETER	DESCRIPTION
AutostartResources=`<App/Desktop1;App/Desktop2;...>`	List of Citrix applications/desktops to be started automatically after logon (Currently only available with `PreLogin=true`)

`[Environment_Store<1-N>]`.

PARAMETER	DESCRIPTION
Url=`<Store-URL>`	URL of the Citrix store
FriendlyName=`<Display name>`	Display name for the Citrix store
Default=true|`false`	Determines whether this store is displayed as default store
Environment=`<Group name>`	Specifies the group by which the stores are grouped (freely definable character string)
AutostartResources=`<App/Desktop1;App/Desktop2;...>`	List of Citrix applications/desktops to start automatically after logon (Currently only available with `PreLogin=true`)

4. To change the design of the Citrix dialogs for all Citrix connections, in the `kioskmode.ini` file, add a section named `[Theme]`:

PARAMETER	DESCRIPTION
ThemeName=`<Themes>`	Name of the Citrix theme. Default: `ucselfservice`
Decorated=true|`false`	Determines whether the windows are shown with window decoration
ColorHover=`<RGB color code>`	Background color for list elements on mouse hover (Citrix list selection widgets). Default: `#e6f1f7`
ColorUnselected=`<RGB color code>`	Background color for unselected list items (Citrix list selection widgets). Default: `#ffffff`

PARAMETER	DESCRIPTION
ColorSelected=`<RGB color code>`	Background color for selected list elements (Citrix list selection widgets). Default: `#cce3f0`

5. Transfer the `kioskmode.ini` file to the devices to `/setup/kioskmode.ini`. To do so, use the Scout feature **Files configured for transfer**. For further information, see Files configured for transfer.

The `terminal.ini` entry along with the `kioskmode.ini` file on a device will cause the device to open the Citrix Self-Service interface in kiosk mode after start-up. All configured stores are available to the user. No other applications can be started apart from the self-service interface.

> **Note:**
>
> If more than one store is configured, users can switch between the stores by clicking the **Store** button on the taskbar while pressing `Shift`.

EXAMPLE FOR

```
[Parameters]

ReconnectOnLogon=true

ReconnectOnLaunchOrRefresh=true

SharedUserMode=true

SelfSelection=false

ShowLastUser=true

LogoffMode=3

LogoffDelay=10



[Store1]
```

```
Url=<https://xd7a/int.sampletec-01.com/Citrix/Store/discovery>

FriendlyName=XenApp A



[Environment_Store1]

Url=<https://xd7b./int.sampletec-01.com/Citrix/Store/discovery>

FriendlyName=XenApp B

Default=true

Environment=PROD



[Environment_Store2]

Url=<https://xd7c./int.sampletec-01.com/Citrix/Store/discovery>

FriendlyName=XenApp C

Default=false

Environment=INT



[Theme]

ThemeName=ucselfservice

Decorated=false

ColorHover=#b0b0b0

ColorUnselected=#a0a0a0

ColorSelected=#c0c0c0
```

Browser in kiosk mode (current eLux® versions)

For kiosk mode, the Builtin Browser is used. In kiosk mode, the browser is started in full-screen mode and with limited user rights. The user cannot open other windows and cannot exit the browser. Note that the Builtin Browser if defined as a browser application is not run in kiosk mode even if address and navigation bar are hidden.

> **Note:**

> Firefox could be run in kiosk mode up to version ESR 52.8. Mozilla supports kiosk mode again starting with version 71.0, but without configuration options. For further information, see Kiosk mode for Firefox.

> Defining Builtin Browser in kiosk mode[1[Kiosk mode via console (GUI) will be supported by future eLux versions]]

1. For the relevant devices, open and define the following entry:

File	`/setup/terminal.ini`
Section	Layout
Entry	`KioskMode`
Value	2

   For further information, see Advanced file entries.

   This parameter enables the kiosk mode for the browser application.

2. Create a text file named `kioskmode.ini` and add the section header `[Browser1]`.
3. Below, enter the relevant parameters:[2[Default values are displayed in bold]]

PARAMETER	DESCRIPTION
Url	Web page (URL) that opens after the browser is started

PARAMETER	DESCRIPTION
Homepage=`<URL of homepage>`	Web page (URL) that opens when users click **Home**
SaveFirstLink = `true\false`	If **true**, the first URL loaded when coming from the startup page will be saved as the new startup page.
Navbar=`true\false`	The navigation bar will be shown.
NavbarPrint=`true\false`	The **Print** button will be shown on the navigation bar.
NavbarForward=`true\false`	The **Forward** button will be shown on the navigation bar.
NavbarBackward=`true\false`	The **Backward** button will be shown on the navigation bar.
NavbarHome=`true\false`	The **Home** button will be shown on the navigation bar.
NavbarUrl=`true\false`	The address bar will be shown.
NavbarRefresh=`true\false`	The **Refresh** button will be shown on the navigation bar.
Favorite=`true\false`	If **true**, the object is marked as a favorite.
FriendlyName = `<Anzeigename>`	Display name for the user

4. Transfer the `kioskmode.ini` file to the devices to `/setup/kioskmode.ini`. To do so, use the Scout feature **Files configured for transfer**. For further information, see Files configured for transfer.

The `terminal.ini` entry along with the `kioskmode.ini` file on the device will cause the Builtin-Browser to open in kiosk mode after system start-up. The configured functions are available to the user. No applications other than the browser can be started.

Note:

> For correct use of the kiosk mode, we recommend that you restrict the user rights for the device as required.

DEFINING DIFFERENT WEB PAGES FOR DEVICES/OUS

If you want your devices to start with different browser startup pages (`Url`) or have different home pages (`Homepage`), you can parameterize the web pages using environment variables. Carry out the steps described above with the following differences:

1. In the `kioskmode.ini`, in the section, for the first and/or second value, set a variable. Example: `Url=$URL1` `Homepage=$URL2`
2. Define the variables used in the `kioskmode.ini` as environment variables for the relevant devices (Advanced device configuration). Example: `URL1=https://www.unicon.com` `URL2=https://myelux.com`

   For further information, see Environment variables.

Local web sites

from eLux® RP 6 2110

On the devices, browser applications can be configured to run without a network connection. To do so, the required HTML pages and scripts are transferred to the devices and the Builtin browser is configured accordingly.This allows you to provide users with a web application that starts automatically and serves as an entry point.

- Define all allowed links from here as HTML links in your scripts.
- Configure all allowed actions via the browser properties.

Combined with kiosk mode, users are restricted to a specific start application and use it as a kind of landing page.

Providing local web sites

1. Pack all files needed for displaying the local web pages into a file named .
   - Startup page `<start>.html`, example: `index.html`
   - Additional HTML pages and Javascript files (optional)
   - Stylesheet (optional)

> **Note:**

> The file name for the archive is predefined and case-sensitive `(landingPage.zip)`. The archive with this name is also required for a single HTML page.

2. Transfer the file to the devices to To do so, use the **Files configured for transfer** feature. For further information, see Files configured for transfer.

> **Note:**
>
> Note case-sensitivity (landingPage.zip).
>
> The next time the Builtin browser is launched on the devices, the archive will be unpacked to `/tmp/browser/landingPage/`

3. To make the web pages available in kiosk mode, enable and configure the kiosk mode. When doing so, set the browser startup page to `file:///tmp/browser/landingPage/<start>.html`

   For further information, see Applications in kiosk mode.

4. To make the web pages available without kiosk mode, define a browser application with the following properties:

   - Browser type: `Builtin`
   - Startup page: `file:///tmp/browser/landingPage/<start>.html`

   Optionally, select **Start Automatically** and, under **Advanced**, specify navigation elements you want to display

At the next browser start (if configured: automatically after device restart) the startup page is displayed. From here, users can connect to the link destinations you have configured via the local web pages.

Even in kiosk mode, you are free to define multiple browser applications. Each browser application has its individual startup page, and one of the browser applications is defined as the default. Users can then switch between the applications using the selector on the system bar.

Troubleshooting application definition

ERROR / PROBLEM	REASON	SOLUTION
Missing firmware	The required software is not installed on the device	Install the software on the device. For further information, see Creating an IDF in the ELIAS guide and Firmware update.
Doubled names	Two applications have the same name. This causes conflicts because applications are identified by their names.	Use unique names.
Hidden application cannot be executed	Applications are invisible for the user when they run in hidden mode. This option is available for applications of the **custom** type.	Enable the option **Start automatically** or **Application restart** to start hidden applications on start or to run them non-stop, respectively.
Problems with certificates in combination with [VMware] server	Server problem occurred	Create a server certificate in the **Windows-CA** with **FQDN**.
	After successful installation, the VMware server uses a self-signed certificate. If a device is configured correctly, it will not accept. The reason is that the **FQDN** (fully qualified domain name) is mandatory for server certificates.	If you use `mmc`: Create a server certificate using the Snap-In `Certificates (Local computer)`.
		The key must be exportable.
		The display name of the server must be `vdm`. The name must be unique in the certificate store **Local computer / Personal**.
COM port redirection in RDP session does not work	Communication errors such as high latencies in the network between your serial device and the virtual desktop	Use the **permissive** mode for the RDP application. This parameter causes communication errors to be downgraded to warnings, and

ERROR / PROBLEM	REASON	SOLUTION
	do not allow serial communication.	communication becomes more tolerant of timeouts.
		Define a free parameter in your RDP application definition with the **permissive** option.
		Example
		`FreeRDPParams=/serial:COM1,/dev/ttyS0,Serial,permissive`
		For further information, see Defining free application parameters.

Third party software

To install additional applications on the devices, carry out the following steps:

1. From our myelux.com portal, for the relevant eLux version, download the specified software package.
2. In ELIAS, import the software package into your container. For further information, see Importing packages into a container in the **ELIAS 18** guide.
3. In ELIAS, add the package to your image in the **ELIAS 18** guide.
4. For the relevant devices, perform a firmware update to the modified image. For further information, see Firmware update.

   The software package is installed on the devices.
5. Configure the software in the back-end environment.

- Avaya Equinox
- Cisco JVDI
- deviceTRUST
- DriveLock
- Grundig Citrix Extensions

- HDX RealTime Media Engine
- JabraXpress Device Updater
- Microsoft Teams
- Nutanix Frame
- Olympus Dictation
- Philips Speech
- SecMaker
- Zoom

Avaya Equinox

Soft phone application providing access to Unified Communications (UC) and Over the Top (OTT) services

Package name: **Avaya Equinox VDI Client**

Cisco JVDI

Jabber Softphone for VDI (JVDI), formerly VXME, extends the Cisco collaboration experience to virtual deployments. With supported versions of Cisco Jabber for Windows, users can send and receive phone calls on their hosted virtual desktops (HVD). The JVDI software routes all audio and video streams directly from one device to another, or to a phone, without going through the HVD.

Package names: Cisco JVDI Client and **Utilities for Cisco JVDI Client**

- Follow the **Cisco Deployment and Installation Workflow** on the Cisco website in order to configure the JVDI system environment.

deviceTRUST®

Dynamic context awareness, allowing users to access their corporate workspace from any location on any device while meeting IT governance requirements

Package name: **deviceTRUST Contextualizing IT**

DriveLock

DriveLock provides endpoint security for USB interfaces on the devices.

Package name: **DriveLock**

Grundig Citrix Extensions

Digital dictation solution by Grundig Business Systems

Package name: **Grundig Citrix Extensions**

HDX™ RealTime Media Engine

The HDX RealTime Media Engine (RTME) enables better audio and video quality for VOIP and video chat.

Package name: **Citrix HDX RTME**

- Configure Microsoft Lync or Skype for Business in the back-end environment.

JabraXpress Device Updater

Solution installed on an end user's device that governs Jabra device configurations such as firmware versions and device settings

A detailed description of how to use the Jabra Xpress Device Updater can be found in the documentation of Jabra Xpress for Linux: `<https://jabraxpress.jabra.com/Downloadables/Linux/UserGuide.pdf>`

Package name: **JabraXpress Device Updater**

Microsoft Teams

Enterprise video conferencing with real-time messaging and content sharing

Package name: **Microsoft Teams Client**

Microsoft Teams Client for Linux is a local application on eLux.

Parameter for the application definition: `teams`

Nutanix Frame

Cloud- hosted Desktop as a Service (DaaS) that empowers any organization to deliver and manage their desktops via a single console for seamless control and administration, providing a true hybrid experience.

Package name: **Nutanix Frame Client**

Nutanix Frame Client for Linux is a local application on eLux.

Parameter for the application definition: `Frame` Parameter with URL: `Frame -url=console.nutanix.com`

Olympus Dictation

Digital dictation solution by Olympus

Package name: **Olympus Dictation Drivers**

Philips Speech

Digital dictation solution from Philips Speech Processing Solutions

Package name: **Philips Speech Drivers**

SecMaker

SecMaker Net iD Enterprise is a middleware supporting SSL 3.0/TLS 1.0 (client identification), PKCS #7 (digital signatures), and PKCS #10 (certificate requests). To use SecMaker Net iD 6.x on the device, a separate license is required.

Package name:**SecMaker NetID**

- Use the Scout feature **Files** to transfer the license file to the devices to `/setup/iid/netidlicense.lic`.

For further information, see Files configured for transfer in the **Scout** guide.

> **Note:**

> The license agreement (EULA) is available on the device under `/etc/iid/SecMaker License and Support Conditions 20150202.pdf`

Zoom

Enterprise video conferencing with real-time messaging and content sharing

Package name: **Zoom Client for Linux**

Zoom Client for Linux is a local application on eLux® that enables peer-to-peer connections between connected devices.

Package name: **Zoom Citrix Plugin**

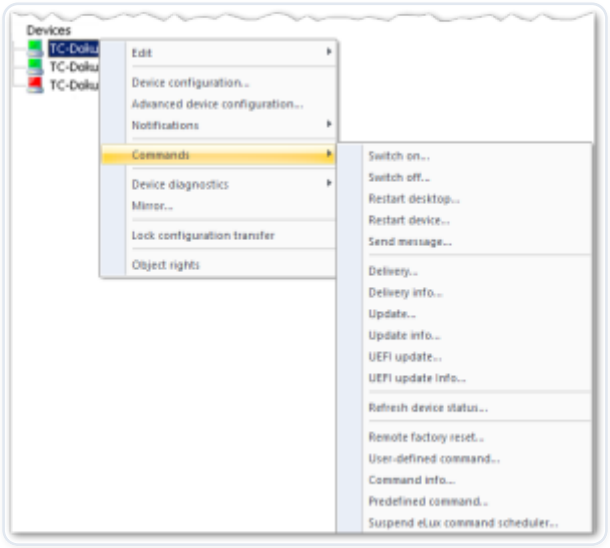
With the Zoom Citrix plugin, ZoomVDI can be used in a virtual desktop infrastructure with a Citrix solution. Peer-to-peer connections are not supported in this scenario.

Client remote management by commands

Administrators can use Scout commands to change the status of the devices, perform updates and send messages. The commands can be executed immediately or can be scheduled to be run once or periodically.

You can apply Scout commands on individual devices, on OUs and on Dynamic Device Groups.

In addition, the context menu of an individual device provides the commands for device diagnostics and for mirroring.



By default, before Scout commands are sent to the selected devices, their current device status is determined. Devices that are switched off are then switched on, if possible, and after command execution are switched off again.

In contrast to commands processed by the eLux® Command Scheduler, commands cannot be "stored" and executed at a later time when a device is up again.

- Available commands
- Executing commands
- Scheduling commands
- Command options
- Command results and update information per device
- Command history
- Factory reset command
- User-defined Commands
- Creating predefined commands

- Defining templates for standard commands
- eLux Command Scheduler

Available commands

The context menu of a device, OU or Dynamic Client Group provides the following **Command** options resulting in the **Execute/Schedule command** dialog for further configuration:

COMMAND	DESCRIPTION
Switch on	Switches on the devices
Switch off	Switches off the devices
Restart desktop	Restarts the eLux® interface.
Restart device	Restarts the devices
Send message	Sends a message to the devices
Delivery	Delivers software for a firmware update
Update	Performs a firmware update
UEFI update	Performs an update of the UEFI firmware
	For further information, see UEFI update in analogy to firmware update in the `UEFI/BIOS` update guide.
Refresh device status	Requests the current device status and refreshes the status of the devices in the tree view
Remote factory reset	Sets the devices back to initial state
	The configuration is deleted, the IDF remains.
	For further information, see Factory reset command.
User-defined command	Enter a user-defined command that will be sent to the devices.

COMMAND	DESCRIPTION
	For further information, see User-defined Commands.
Predefined command	Provides user-defined commands that have been predefined globally. For further information, see Creating predefined commands.
Suspend eLux command scheduler	If recurring eLux commands that are scheduled locally on the device by the eLux Command Scheduler (cron jobs) have been defined by the administrator, the execution of these commands is temporarily stopped until the next restart of the device.
Configuration run. (not available for individual devices)	Prepares the configuration data for an OU or Dynamic Client Group. For further information, see Configuration run. This command is not available for an individual device.

For further information on options you can add to the respective commands see Command options.

The following options open the relevant log file:

COMMAND	DESCRIPTION
Delivery Info	Opens the log file of the latest software delivery
Update Info	Opens the log file of the latest firmware update
Command Info	Opens the log file of the latest user-defined command

Command options

The options you can use to execute a command vary to some extent depending on the selected command.

General and common options

OPTION	DESCRIPTION
Command	The selected command is displayed. , The list-field allows switching to other commands.
Inform user	Users are informed by a notification before the command is executed. , Specify a time period in seconds for displaying the notification. The command will be executed after the time period has expired. , If the time period is set to [0], the notification will be displayed and the system waits until the user confirms the command execution. , For firmware updates or UEFI updates, additional deferment options can be configured in the device configuration. For further information, see Update deferment by users.
User can cancel	Users are allowed to prevent the command execution.
Send only to devices with **Switched-on** status [from Scout 15 2204]	Only for **Update, Delivery, User-defined and Pre-defined commands** , The command is sent only to devices shown with **Switched-on** status. Switched-off devices are skipped. The actual device status is not determined before command execution. , If the **Check reachability** option is also selected, a **ping** command is used to determine the current status of the relevant devices before the command is issued. The command is only sent to the responding devices. This avoids timeouts that can be caused by devices that are not ready for operation despite showing a **Switched-on** status. , Default (both options disabled): The device status of the relevant devices is determined before the command is executed. If possible, switched-off devices are switched on for command execution and then switched off again. If a large number of devices is affected, we recommend that you select the

OPTION	DESCRIPTION
	above options instead. This allows for a faster and smoother operation.
Run with system rights	Some commands require system rights which are checked before execution.
Now/Once/Every	Time of execution, see Scheduling commands
On more devices wait	Delay command execution between individual devices
Include sub-OUs	The command is additionally applied to all devices in subordinate OUs.

Command-specific options

OPTION	DESCRIPTION
Send message /	Select the message to send from the list or enter your message text. To format your message text, use HTML tags.
Message	Optionally edit the title of the message.
Title	Optionally specify the display duration in seconds. With 0 seconds, the system message will be shown until the user clicks **OK**.
Visible	
	Selected or all previously sent message texts can be deleted from the list via button.
Delivery / Clean update partition before delivery	Before writing to the flash memory, the update partition is cleaned up.
	All files of the current software image must be re-transferred from the web server. For further information, see Performing deliveries via command.
Update / Format system partition before update	Before the software is installed, the system partition of the flash memory is formatted.

OPTION	DESCRIPTION
	For the installation of the firmware update, all files of the current software image must be re-transferred from the web server. For further information, see Performing updates via command.
UEFI update / Overwrite identical or higher version	The UEFI system is always overwritten. This is necessary for a downgrade, for example. For further information see Performing UEFI updates via Scout command.
Remote factory reset /	See Factory reset command
Retain local configuration	
Delete **Scout Server** address	
User-defined command	Below the command name, select a command from the list or enter a command. For further information see User-defined Commands.
Predefined command / Selection	Select a predefined command from the list. The function requires that commands have been predefined centrally, see Creating predefined commands.

Command results and update information per device

Feedback on performed Update, Delivery and User-defined commands is available for individual devices

- in the **Properties** window
- in the **Update info/Command info/UEFI update info** dialog (via device context menu)

> **Note:**

> In addition, all executed commands for all devices are displayed in the **Command history** window. For further information, see Command history.

> All processes are recorded, even if the commands turn out to be obsolete and haven't been run or have been aborted. Successfully completed commands are shown with a green symbol.

Viewing command results for an individual device

The following instructions are related to **Update** commands. Viewing results of a UEFI update, delivery or user-defined command is done accordingly.

1. To show the **Properties** window, click **View > Window > Properties**.

   The **Properties** window is shown permanently in the upper right. For the selected device some properties are shown. Properties can be shown or hidden by using the icon.

2. Select the relevant device in the tree view.

   In the **Properties** window, in the Update section, the following fields are provided:

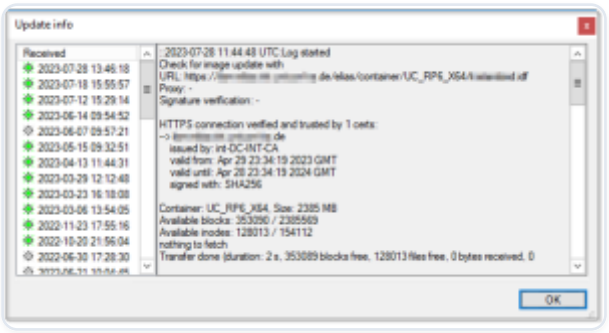
Image	Current image
Update time	Exact point in time of the latest update
Update status	Current status such as , or
Update provider	Origin of software packages (web server or proxy)
Update size	Size of the transferred packages in compressed format

> **Note:**
>
> The properties **Update provider** and **Update size** are evaluated for updates, but not for the migration to a major version or a downgrade.

3. Double-click the term **Update status** or click ... at the end of the line.

   The **Update Info** window is displayed. On the left, you can see all updates that have been processed, aborted or not processed because the image had been up-to-date. For a selected

update command, you can view all logged data on the right side, among them the installed software packages.



Command Info

Information on the last command or update of a device can also be viewed by using the device context menu in the tree view.

- From the context menu, choose **Commands > Commando Info** or **Delivery Info, Update Info, UEFI update Info**.

Scout Report Generator

Use the following fields for your evaluations:

Command name	Executed command for user-defined commands
Command time	Time stamp of command execution
(UEFI) update time	
Delivery time	
Command result	Successful / Failed Not available
Update state	
Delivery state	Further values are available for Update and Delivery commands.
Command output[1[[ ]from Scout 15 2204]]	Only for user-defined commands with specific results

	Note that the **Command result** may be successful for many commands while there is no **Command output**.

For Update, UEFI update and Delivery commands, further fields are provided corresponding to the fields in the **Properties** window.

Command history

All of the executed **Update, Delivery, and User-defined** commands can be viewed in the command history. When calling the command history, the object rights of the administrator management are respected.

- Click **View > Command history**

The **Command history** window opens and displays one job (command for 1 to n devices) per line providing the following information:

OPTION	DESCRIPTION
Type	Type of object the command is applied to. This can be an individual device, an OU with sub units (OU+), an OU without sub units (OU) or a Dynamic Client Group.
Name	Object name (name of device, OU or Dynamic Client Group)
Command	Executed command (Update, Delivery or User-defined command)
Devices	Number of devices concerned
Start	Date and time of sending command to the devices / starting time
End	Date and time of sending command to the devices / ending time
	The ending time of a job is reached when either the devices report back `Successful` or `Failed` or when the timeout of 5 minutes

OPTION	DESCRIPTION
	for feedback is passed. If the administrator terminates a job, the ending time is defined by the terminating time.
Successful	Number of devices that have successfully processed the command
Failed	Number of devices that have reported failure during command processing
Timeout	Number of devices that haven't reported feedback within the defined time period of 5 minutes
Progress %	Progress of command processing in percent, across all concerned devices
Administrator	Administrator who ran the command

Apply the following options to the job list:

OPTION	ACTION
Refresh	Press `F5`.
Sort table rows	Click the column title by which you want to sort.
	A first click sorts the jobs ascending, and the second one sorts descending.
	To reproduce the default sorting order, click `F5`.

Apply the following options to a selected job:

OPTION	ACTION
View details	Click **Details**.

OPTION	ACTION
	The **Command details** window displays all processing details of the concerned devices. Along with starting and ending times you can find the current status and the command processing result for each device.
Search object in **Scout** tree view	Right-click an object name, and then click **Find in tree view**.
	The first result is selected in the tree view.
Terminate running job	Select the running job, and then click **Terminate**.
	A command terminating request is sent to the **Scout Server** and the transmission of the command to the devices is stopped.

Factory reset command

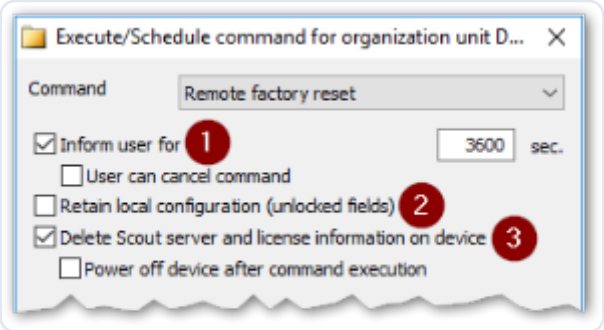
Use the **Remote factory reset** command to reset the configuration of the relevant devices to their initial state. The local device configuration and local application definitions will be deleted.

The following data are retained:

- The installed image with all software packages (firmware)
- Connection data to the Scout Server including the server address and OU ID (default)
- License information (default)

After the device has received the command, the current device configuration and local application definitions are deleted. By default, after a restart, the device reconnects to its and obtains the configuration of the OU to which it is assigned.

Additional options of the Factory reset command



1	Users are informed before command execution, see Command options.
2	Prevent local device configuration from being overridden
3	All configuration data on the device will be deleted. The device then loses connection to its Scout Server. Optionally define to switch off the device afterwards.

OPTION	DESCRIPTION
Retain local configuration (unlocked fields)	User-defined values of the local device configuration in unlocked fields are retained. This only applies to fields that the user is allowed to edit. , -- only available if allowed in **Advanced options > Devices** -- , For further information, see Supporting local configuration.
Delete Scout Server and license information on device	In addition to the configuration data, the following data are deleted
	• Address of the Scout Server machine
	• License information stored on device (license lease) from Scout 15 2302 (previously available as a separate

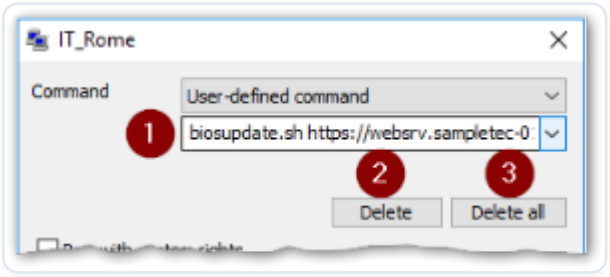
OPTION	DESCRIPTION
	option) , This option can be used for the resale of devices, for example.
	• Certificates in [/setup/cacerts]
	• WLAN configuration
	• 802.1X configuration
	• Private key of SCEP client certificate stored in a TPM 2.0 module
	An encrypted setup partition (TPM 2.0) will be decrypted. , If you perform the **Remote factory reset** command with this option selected, it corresponds to a local **Factory reset** triggered by the relevant eLux command in the eLux **Command panel** on the device. For further information, see Device management in the **eLux** guide.
Power off device after command execution[2[[ ]from Scout 15 2302 (previously done automatically after deleting the license information)]]	-- only available with **Delete Scout Server and license information on device** --
	The device is shut down and remains switched off.

Note:

The **Remote factory reset** affects the device's setup partition. The system partition with the installed firmware, in contrast, is `not` affected by the factory reset and is only changed using the **Update** command with the option **Format system partition before update**.

User-defined Commands

Authorized administrators may issue individual commands to the devices. A user-defined command can be executed with system rights so that far-reaching operations can be performed. The correct syntax is particularly important here.



1. Select a command from the list (recently used commands) or enter it.
2. Delete the selected command.
3. Delete all user-defined commands.

> **Important:**

> Always test a user-defined command in a test environment before using it productively.

> User-defined commands can be helpful for tasks such as the following:

- Perform a special UEFI/BIOS update
- Update a license lease
- Wipe data disks

> **Note:**

> After having executed a user-defined command, after a time span of 30 seconds, you can run the next user-defined command or update command.

- Wiping the data disk

Wiping the data disk

-- from --

To permanently erase all data of devices from their flash memory, a script is available that you can apply as a user-defined command. The script overwrites all partitions of the primary disk with zeros.

1. Make sure that the devices have the required operating system version.
2. For the relevant device, OU or Dynamic Client Group, from the context menu, choose **Commands > User-defined command**.
3. To run the script, below **User-defined command**, enter the following command: **wipe-device really-wipe**

   The script must be run with the **really-wipe** parameter to prevent accidental execution and wiping of the disk.
4. Select **Run with system rights**.
5. Click **Execute**.

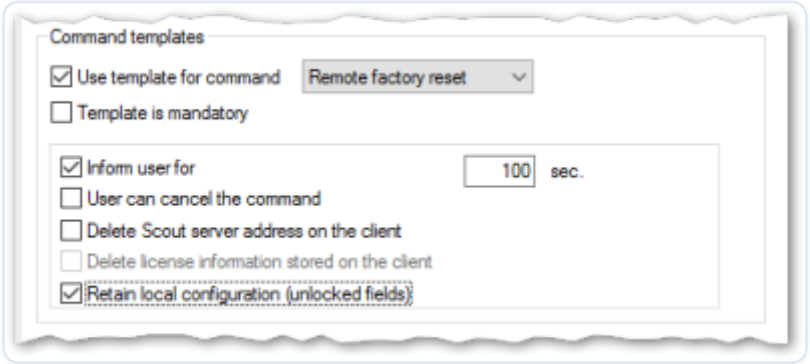
The relevant devices confirm the command execution (last contact to the Scout Server) and shut down. The devices then restart one more time with the core installer, which zeros all partitions of the disk. After that, the devices shut down finally.

Defining templates for standard commands

For standard commands such as **Restart** or **Update**, you can define preferences via templates.

1. On the Scout menu, click **Options > Advanced options > Predefined commands**.
2. Under **Command templates**, select **Use template for command** and then, in the list-field, select the relevant command.
3. To define the preferences as mandatory, select **Template is mandatory**.

   If a template is not mandatory, the operative administrators can overwrite the specified default values.
4. Specify the values for the selected command.



A template may contain command-specific preferences as well as general preferences such as

- Inform user for x seconds
- User can cancel command

> **Note:**
>
> Templates for standard commands are only available in the Scout Console but not in the Scout Dashboard.

eLux® Command Scheduler

-- from Scout 15 2103 --

Recurring commands can be defined so that they are scheduled and executed locally on the device by the eLux Command Scheduler. Like cron jobs, these commands are initiated by the device at defined points in time, whereas Scout commands are initiated on the server side. The time of execution depends on the time zone in which the device is located.

The eLux Command Scheduler can process the following commands:

- Update firmware
- Synchronize device configuration
- Restart device
- Shut down device
- User-defined command

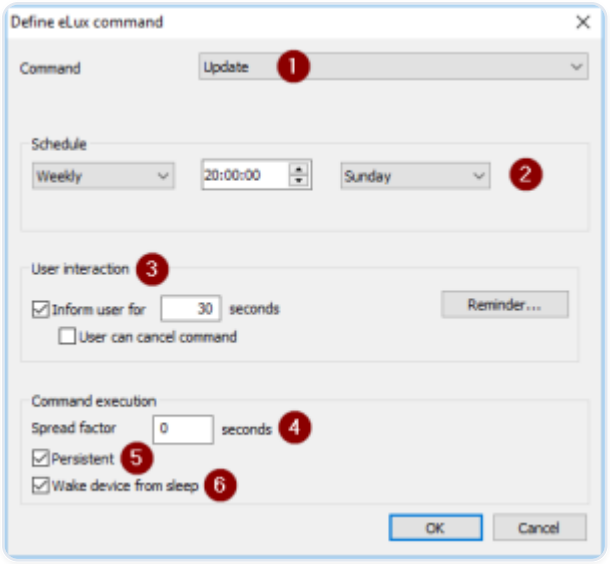
The commands for the eLux Command Scheduler may be defined either on the top level in the **Advanced options** or in the **Advanced device configuration** of an OU. For each OU, you can define whether you want the jobs of the eLux Command Scheduler defined and scheduled above to be applied or not.

- Defining eLux Commands
- Deferment options for users
- Suspending eLux Command Scheduler
- Command definition parameters

Defining eLux® Commands

-- from --

The eLux Command Scheduler is configured in the Scout Console.



1	Select command
2	Configure periodic scheduling
3	Configure user interaction and deferment options
4	Generate random delays between devices
5	Missed jobs are caught up
6	Devices are woken up from suspend to execute jobs

1. Open **Options > Advanced options** or the **Advanced device configuration** of an OU. Then choose the **eLux Command Scheduler** tab.
2. To define a new command, click **Add**.
3. In the **Define eLux Command** dialog, edit the following fields:

OPTION	DESCRIPTION
Command	Type of command

OPTION	DESCRIPTION
	To execute **user-defined commands**, system rights are required. Therefore, include the password.
Schedule	Periodic scheduling of the command
	If the provided options are still too few, in the , you can define a string using the systemd syntax. For further information, see Command definition parameters.
User interaction	Determine whether and how long you want to inform the users before the command is executed, and whether they are allowed to defer it. For further information, see Deferment options for users.
Spread factor	Time span in which random delays are generated to prevent simultaneous execution on many devices
	Example 600 seconds: If the execution time is 6:00 pm, a randomly generated time value for the execution of the command per device, which may be a maximum of 600 seconds, is added to the time 6:00 pm. The command is therefore executed between 6:00 and 6:10 for all devices.
	Default:
Persistent	Missed command executions (jobs) are immediately caught up after the next device start. If multiple repetitions of the same command were scheduled, the command is repeated once.
Wake device from sleep	Devices will be woken up from sleep mode (suspend) to execute scheduled commands.

4. Confirm with **OK** and **Apply**.

The scheduled command is displayed in the **eLux Command Scheduler** tab of the **Advanced Options** and in the **eLux Command Scheduler** tab of the **Advanced Device Configuration** for all OUs. Creating and editing is only allowed in the **Advanced Options**.

5. Define which OUs you want (not) to inherit the jobs of the eLux Command Scheduler. By default, the settings (jobs) in are inherited from the next higher instance.

The scheduled command is executed by all devices whose OU inherits the jobs of the eLux Command Scheduler at the next due point in time.

> **Note:**
>
> To view information on the status and the next execution of a command, on the device, use `systemctl status <[jobId]>.service` and `systemctl status <[jobId]>.timer`.

Deferment options for users

Similar to Scout commands you can also define for the command definitions of the eLux® Command Scheduler whether the users are informed before the command is executed and whether they are allowed to cancel or defer it.

OPTION	DESCRIPTION
Inform user	The user is informed by a notification before the command is executed.
	Specify the time period in seconds for displaying the notification. The command is subsequently executed.
	If the value is 0, the notification will be displayed until the user confirms the execution of the command via button.
User can cancel	The user is allowed to prevent the execution of the command via button.
Reminder > Number of allowed deferments	Define how often the user is allowed to postpone.

OPTION	DESCRIPTION
Reminder > Delays until next reminder	Select one or more time periods users can choose to defer.

For further information, see also User information before update

Suspending eLux® Command Scheduler

The recurring execution of defined commands by the eLux Command Scheduler can be temporarily suspended from the . To do so, the administrator sends a command to the relevant devices, which stops the eLux Command Scheduler until the next restart of the devices. After the next device restart, the eLux Command Scheduler continues to process the scheduled commands as planned.

1. For the relevant device, OU or Dynamic Client Group, from the context menu, choose **Commands > Suspend eLux Command Scheduler**.
2. Click **Execute**.

The execution of all commands scheduled by the eLux Command Scheduler is suspended with immediate effect until the next restart of the devices.

> **Note:**
>
> If commands are defined with `Persistent=true` and the device is switched off at the time of scheduled command execution, these commands (persistent jobs) are started subsequently after the next restart.

Command definition parameters

For each command definition, in the `scheduler.ini` file, a new `Job<N>` section is created. Find all command definition parameters with their possible values below.

PARAMETER	DESCRIPTION
Id	Character string for unique identification of the command
Type	Command type

PARAMETER	DESCRIPTION
	1 Update
	2 Synchronize configuration
	3 Restart device
	4[1[[ ]from Scout 15 2103]] User-defined
	5 Shut down
Schedule	The format follows the `systemd calendar events`. Syntax for a timestamp
	Tue 2020-01-21 11:12:13
	Minutely [--* ::00]
	Hourly [---:00:00]
	Daily [--* 00:00:00]
	Monthly [--01 00:00:00]
	Weekly [Mon --* 00:00:00]
	Examples
	[--* 4:00:00] Every day at 4:00 am
	[Mon..Fri --* 22:30] Every workday at 10:30 pm
	[-*-1,5 11:12] Every first and fifth day of any month at 11:12. am
	[Sun 2020-- 17:15] Every Sunday of the year 2020 at 5:15 pm
	For further information, see https://www.freedesktop.org/software/systemd/man/systemd.time.html

PARAMETER	DESCRIPTION
SpreadFactor	Time span in which random delays are generated to prevent simultaneous execution on many devices
	Default: [0]
Persistent	`true`[false]
	If [true], missed command executions (jobs) are immediately recovered after the next device start.
Command	Only for [Type=4]: Command for user-defined command
WakeSystem	true`false`
	If [true], the system will be woken up from sleep mode (suspend) to execute scheduled commands.

Remote maintenance

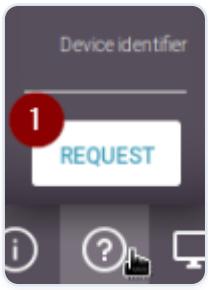
For maintenance, user help-desk and troubleshooting purposes, the administrator can use different tools to access the devices.

- Device identifier for support
- Mirroring
- Device diagnostics

Device identifier for support

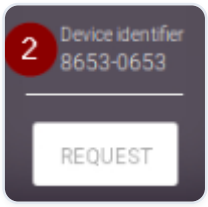
From Scout 15 2101 and eLux® RP 6 2101

To provide support for users, the first step is to identify the relevant device. As an alternative to the IP address or MAC address in the **Information** dialog, eLux provides a temporary device identifier that users can easily request.



Users have a question mark icon on the system bar, provided the user right **Request simple device identifier** is enabled (default).

- To request the so-called **Simple device identifier**, users click (or right-click) the question mark icon and then **Request** (1).



The system generates and transmits a temporary device identifier for the requesting device (2).

The user gives this device identifier to the support staff or administrator. By default, the device identifier has a validity of five minutes.

The validity time and optional additional text to be displayed can be specified by the administrator in the Scout Console under **Options > Advanced options > Rules**.

Device diagnostics

> **Note:**

> This feature can only be applied to an individual device.

> Device diagnostics help you run predefined commands on the device and retrieve protocol and configuration files from the device. These are then sent to Scout or other destinations for diagnostic purposes. The requested client files support the administrator in error analysis and are required when you open a support ticket.

Depending on the user rights, the reverse way can be used: Users send diagnostic files or screenshots from the device.

Administrators are free to use the **Request diagnostic files** feature to request freely definable files from a device.

> **Note:**
>
> To compare actual and target device configuration settings of individual devices, use a report. For further information, see Evaluating configuration data.

- Diagnostics scope and content
- Log level
- Requesting diagnostic files
- Defining an individual template
- Further diagnostic adjustments

Diagnostics scope and content

The amount of diagnostic files requested from a device is defined by two factors:

- Selected templates in the **Diagnostic files** dialog In addition to the #System template, further individual templates might be active.
- Configured log level Diagnostics tab Only the enhanced log level allows to execute all scripts on the device and retrieve all files defined in the `#System` template.

SYSTEM TEMPLATE

To perform device diagnostics, a predefined template called `#System` is provided. This template includes a file list with relevant configuration and log files plus a script code to be run on the device. Neither of them can be edited. You cannot deactivate the system template.

The system template contains the following files:

- Log files for eLux® and components
- eLux configuration files, including `terminal.ini, user.ini, sessions.ini`
- Configuration files of standard components and standard applications

The defined script contains different system commands that request software and hardware information from the device. Included are for example runtime information, installed packages, active codecs, the update partition, PCI and USB devices. Via `fwupdmgr`, firmware information about installed components such as the UEFI system or the flash memory is also retrieved.

The output of the system commands is stored in the `mytrace.log` file, which is transmitted along with the other files in a `.zip` archive.

> **Note:**
>
> Authorized administrators may define additional custom templates.

Log level

Two log levels are provided in the device configuration of eLux® devices: **Standard** and **Enhanced**. These levels correspond to switching the enhanced log level **On** and **Off** in the Scout Console.

In normal operation, the standard level is adequate. Before performing device diagnostics, however, temporarily enable the enhanced log level on the device to make sure you retrieve all data needed.

> **Note:**
>
> After the device diagnostics, we recommend that you reset the log level in order not to unnecessarily strain the flash memory capacity of the device.

ENABLING ENHANCED LOGGING

1. Open the device's context menu and click **Device configuration...**.
2. On the **General** tab, clear the option **Use parent device configuration**.
3. On the **Diagnostics** tab, set the **Debug level** option to **On**.
4. Confirm and restart the device.

Enhanced logging for the device is active and you can pull off the diagnostic files.

DISABLING ENHANCED LOGGING

1. Open the device's context menu and click **Device configuration...**.
2. On the **Diagnostics** tab, set the **Debug level** option to `Off`.
3. On the **General** tab, select the option **Use parent device configuration**.
4. Confirm with **Apply** and **OK**.

Enhanced logging for the device is reset to standard and device configuration inheritance is restored.

Defining an individual template

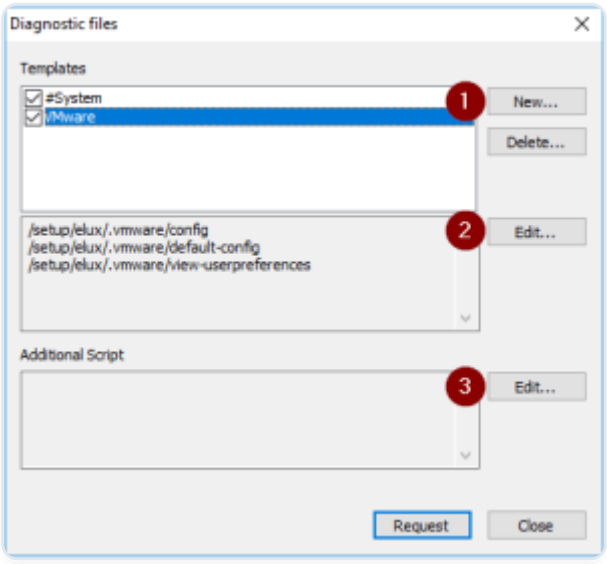
In addition, authorized administrators may define further templates containing files and script. The templates are available globally, no matter where you define them.

REQUIRES
Object right [Edit diagnostic templates] (disabled by default)

1. On the context menu of a device, choose **Device diagnostics > Request files**.

   In the **Edit diagnostic files** dialog, the predefined `#System` template and, if defined, further templates are shown.

2. Click **New** (1). Enter a name for your new template and confirm with **OK**.



3. To define diagnostic files for your template, select the new template and, next to the file list, click **Edit** (2).

   In the text box, enter the relevant file names with path, line by line. Confirm with **Save**.

4. To enter script code you want to run on the device, next to , click **Edit** (3). Authenticate with the device password.[1[from Scout 15 2107]]

   In the text box, enter your code and confirm with **Save**.

> **Note:**

> When you perform the device diagnostics feature with **Request**, all selected templates are included. Whether all of the listed files of the `#System` template are written and transferred, depends on the configured log level.

Further diagnostic adjustments

Diagnostic files over multiple device restarts

REQUIRES
Enhanced logging must be enabled.

When you request diagnostic files, a set of log and configuration files is retrieved and packed on the target device. Included is the file `last_boot.zip`, which contains diagnostic files over a device restart. This `.zip` file contains all files defined for the system template plus some additional ones. You can specify the number of cycles for which the file is created and saved on shutdown of the device. The default is set to five.

The most recent file is named `last_boot.zip.0`. The further back a file was created, the later its sequential number in the file name.

- To define the number of cycles yourself, use the **Advanced file entries** feature with the following parameter:

File	`/setup/terminal.ini`
Section	`Global`
Entry	`LogCycles`
Value	`2-n`, The default value is [5].

> **Note:**

> That the file is only created, if enhanced logging is active. For further information, see Requesting diagnostic files.

STORE DIAGNOSTIC FILES PERSISTENTLY

By default, the diagnostic files are stored on the device under `/setup/logs`.

- To define a different storage location, use the **Advanced file entries** feature with the following parameter:

File	`/setup/terminal.ini`
Section	`Global`
Entry	`LogPath`
Value	`<Directory on the device>` The default is `/setup/logs`
	Example: `/update/logs` A location on the update partition may also be defined.

If you use disk encryption via TPM 2.0, the setup partition will be encrypted. To have the diagnostic files accessible, define the update partition as their storage location. The update partition is then mounted automatically.

DIAGNOSTIC ADJUSTMENTS VIA SOFTWARE PACKAGE

Provide diagnostic settings for new devices you wish to connect before their initial contact with the Scout Server, during installation. To do so, integrate the eLux® software package **Diagnostic adjustments** into the image to be installed.

The **Diagnostic adjustments** package contains two feature packages that you can activate separately:

- **Enhanced logging**: Corresponds to the device configuration **Diagnosis > Debug level** and enables enhanced logging
- **Logs on update partition**: Sets the location for diagnostic files to `/update/logs`

Note that the diagnostic settings set via software package are only effective until the first contact of the device to its Scout Server. Then the device configuration is synchronized and the device receives the configuration data defined for its OU.

Firmware update

On delivery, the are normally equipped with the operating system and the basic software components such as ICA® client, RDP client, browser and emulations. This software called firmware is stored on the flash drive. Whenever new software versions are available or demands change and software components need to be added or removed, the firmware can be updated.

Basic steps

- Download the relevant software packages from our myelux.com portal.
- Modify the image file (IDF) on the web server via ELIAS.
- Check the firmware configuration of the relevant
- Perform the update
- Deliver new software packages
- Install new software packages

To perform the update in one step, use an command. In this case, the required software packages are delivered and then automatically installed. Alternatively, the two actions can be uncoupled: The software is delivered in a first step when you use a command. It is subsequently installed when you run the command.

To save bandwidth, you can use a proxy client for updates. For further information, see Static proxy.

Ways to initiate a firmware update

Updates can be performed immediately or initiated automatically at a defined point in time:

- Use the command feature to execute or schedule (once or periodically) firmware updates.
- Define an . This will result in a firmware update on the next device restart.
- Configure the devices to automatically check for new image versions on start-up or shutdown. If an updated version is available, the update process will be started.

The option is part of the and can be applied to individual devices, OUs and all devices.

If configured, users can defer the execution of a firmware update to a later point in time.

Updates are only performed, when the relevant IDF has been modified. All update activities are logged.

Relevant devices

Commands and notifications can be applied to the following devices and groups:

- Individual devices
- Multiple devices selected in the window (multiple selection via and)
- OUs
-

Recovery-Installation

To reset your devices to initial state, perform a recovery installation. A recovery might also be required if critical feature packages of the Base OS have been changed, or if your operating system has not been updated for a long time. With a recovery installation, all data on the storage medium are wiped (/en-us/unicon-elux-scout/2611/short_guides/recovery_general.html) in the `Recovery` short guide.

Requirements

The following components are required to perform a firmware update:

- Web server (for example IIS) to provide the eLux software packages and image definition files ( ) via HTTP/HTTPS or FTP/FTPS, with the relevant web server role enabled.
- Software container with up-to-date eLux® software packages on the web server
- Image management to create and modify image definition files in a software container: the **Software** section of UniconOS Management Board, see Software, or ELIAS 18
- UniconOS Management Board to configure firmware updates for the devices, see Device configuration / Firmware

Scout Server and UniconOS Management Board are part of the Scout Enterprise Management Suite®. From UniconOS Management 2611, you build images with the image management in UniconOS Management Board. The current software bundle `eLuxversion_AllPackages.zip` and further optional software packages are installed with the eLux container.

Alternatively, install ELIAS 18 and import the eLux software bundle into your container. Then, there is no need for the container installation.

For further information on the installation of the Scout Enterprise Management Suite and the eLux container, see the Installation guide.

Planning firmware updates

Providing a new image

1. Download the latest software packages that are not yet included in your container from our portal. In ELIAS, import them into your container. For further information, see Importing packages into a container in the **ELIAS 18** guide or **Importing software packages** in the ELIAS 18 guide.
2. In ELIAS, add the relevant software packages to your image. Then save the modified image file. For further information, see Creating an IDF in the **ELIAS 18** guide or **Creating an image** in the ELIAS 18 guide.

Checking firmware configuration of the devices

1. For the relevant devices, open the **Device configuration > Firmware**. To apply the update to all devices, choose **Options > Base device configuration**.
2. Check whether the following fields are configured correctly for a firmware update: **Protocol, Server, Path and Image file**.

   The URL shown below the **Path** box is generated based on this data. The URL is relevant for the transfer of the image file `(.idf` and eLux® software packages.

   The specified image file name must be identical to the name of the image file updated in ELIAS.
3. For firmware updates via command: To allow users to postpone the update, configure the reminder settings. User can then control the time of execution themselves. For further information, see Update deferment by users.
4. To update the devices automatically on start or shutdown, select the relevant **Check for new version** option in the bottom area.

   Since in this case the update is initiated by the device, the firmware parameters stored locally on the device are used.
5. Confirm with **OK**.

   For further information on the firmware update configuration, see Configuring firmware updates.

Performing firmware updates

With the appropriate firmware configuration of the devices, you only need to provide an updated image to trigger a firmware update:

As soon as an updated image file is available on the web server and if one of the **Check for new version** options is selected, the update is performed on the next device restart or shutdown.

Note:

If the device is connected via VPN, the **Check for new version** options cannot be used. Use an **Update notification** instead.

Alternatively, you can initiate a firmware update via one of the following procedures:

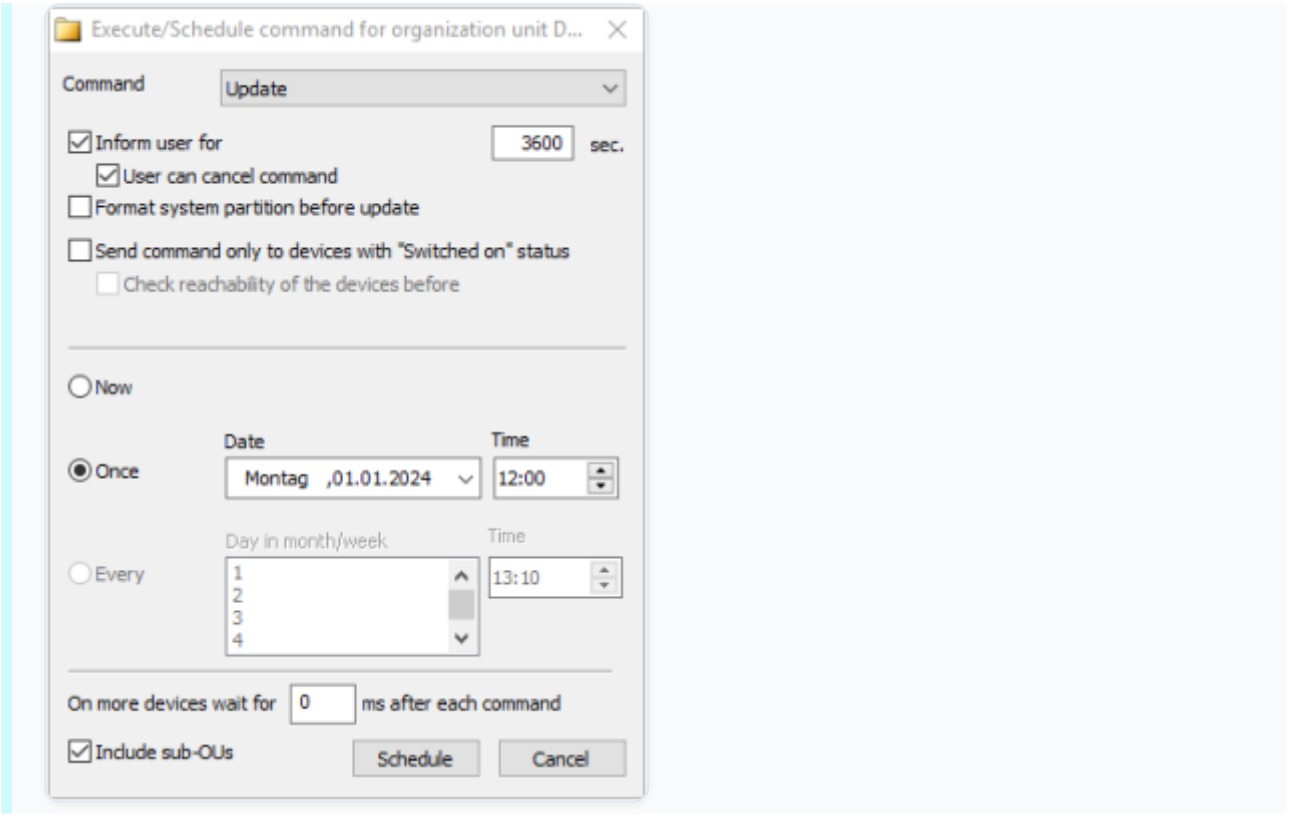
- Perform an update command
- Schedule an update command for a specified time, once or periodically
- Define an update notification

- Performing updates via command
- Updates initiated by the system
- Performing updates via notification

Performing updates via command

Note:

To deliver the software packages in a separate step before performing the update, use the **Delivery** command.



1. Select a device, an OU, a Dynamic Device Group or devices in the **All devices** window.
2. On the context menu, click **Commands > Update**.
3. To inform users before the update will be processed, we recommend that you select the **Inform user** option. For further information, see Command options and Update deferment by users.

   For further information on the impact, see User information before update.
4. To format the system partition of the device's flash memory before writing, select the option **Format system partition before update**.
5. Define the point in time for the update process. For further information, see Scheduling commands and Executing commands.
6. Click **Execute**.

The update process is triggered at the defined time. The update status is displayed for each device in its **Properties** window. During the update process, the status is shown. Detailed information about the currently processed action with time stamp is shown additionally. Example:

Update in progress (Transfer started -- 2022-08-20 11:34:23) Update in progress (Transfer completed -- 2022-08-20 11:35:45) Update in progress (Installation started -- 2022-08-20 11:35:48) Update in progress (Installation completed -- 2022-08-20 11:37:56)

For further information, see Command results and update information per device.

Note that updates are only performed, if the relevant IDF has been modified. If an update fails, no efforts will be made to retry.

> **Note:**
>
> When you execute an **Update** command, the relevant information is transferred to the devices as a URL. To create the URL, the system uses the values set in **Device configuration > Firmware** at the time when the command is run. Note that if the client initiates the update, the local **Firmware** configuration is relevant.

Performing updates via notification

By using update notifications, you can send an explicit one-time update request to selected devices to be evaluated with the next connection. The devices then are updated to the image configured in the Scout firmware configuration.

1. Select a device, an OU, a Dynamic Device Group or devices in the **All devices** window. .
2. On the context menu, click **Notifications > Initiate firmware update**.

   The **Firmware update notification** dialog is shown.
3. Specify whether you want to inform users, and if they are allowed to cancel the command. For further information, see Performing updates via command.
4. To format the system partition before performing the update, select the relevant option.
5. Confirm the notification and confirmation.

   The notifications for firmware updates are defined for the relevant devices.

   For each device, in the **Properties** window, the **Update notification** field shows the value `Activated`.

   If the **Update notification** field in the **Properties** window is hidden, click to define which fields you want to show.

For the relevant devices, a firmware update notification is set. As soon as a device restarts and reconnects to Scout, it receives an update request[1[Note that updates are only performed if the relevant IDF has been modified.]] and the firmware update notification is automatically deleted.

Depending on how you have configured the notification and the device configuration in **Firmware > Reminder settings**, the update is performed immediately or the user receives a system message including deferment options. For further information, see User information before update.

The update status is displayed for each device in its **Properties** window. For further information, see Command results and update information per device. If an update fails, no system-side efforts will be made to retry.

For devices without update partition[2[flash memory less than 4 GB]], an update request might be shown although an update is not required. When the user clicks the **Update** button, the window is closed, no update is initiated.

> **Note:**
>
> In the Scout Report Generator, you can filter devices by the field **Image update notification**.

DELETING THE UPDATE NOTIFICATION FOR ONE OR MORE DEVICES

An update notification can be deleted before the firmware has been updated:

- On the context menu, click **Notifications > Delete update notification**.

> **Note:**
>
> Notifications are always set or deleted for all selected devices regardless of whether they are only available for individual devices.

User information before update

> **Note:**
>
> This feature refers to firmware updates and UEFI updates.[1[from eLux® RP 6 2107]]

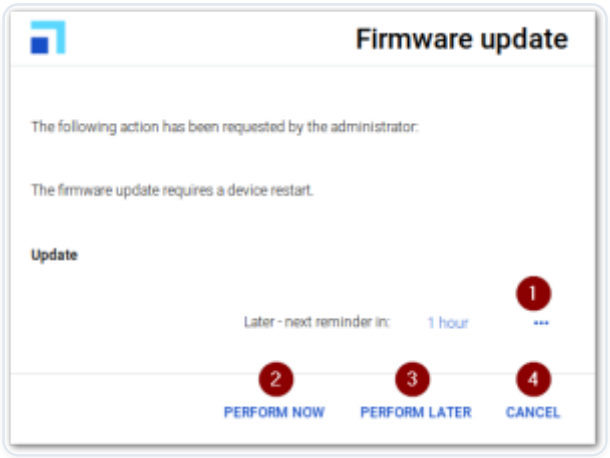
> Before an update is executed, the relevant users can be informed by a system message. Depending on the configuration in **Firmware > Reminder settings**, users are provided with various options for deferring or aborting the request.

The system message is displayed in the following situations:

- The administrator executes an **Update** command with the **Inform user** option.
- In the **Firmware** dialog, the option **Check for new version on start** with user confirmation[2[from Scout 15 2104.3000 and Scout 15 2209 with corresponding eLux versions]] is configured.

While the message is displayed, users may close applications, disconnect sessions, or - if configured - defer the update.

If the display duration for the message has been set to 0, the message will be displayed until the user clicks a button.



Users have the following options:

1	Select a time interval until the next reminder for the firmware update
	The list-field contains the values specified in **Firmware > Reminder > Delays until next reminder**.
	Displayed only, if the **Number of allowed deferments** is set to **1** or higher, and if at least one more deferment is possible.
2	Perform the firmware update immediately
3	Postpone the firmware update by the time period selected (1)

	If the device is shut down before the time interval expires, the update is performed during shutdown.
	Displayed only, if the **Number of allowed deferments** is set to 1 or higher, and if at least one more deferment is possible.
4	Abort the update process definitively
	Displayed only, if the command option **User can cancel the command** is selected.
	There will be no automatic restart of the update process.

Delivering software in advance

Before performing a firmware update, you can deliver the required software packages in a separate step. Only after the software deployment has been reported as successful on all relevant devices, start the installation by using an **Update** command or notification.

- Clients are provided with an update partition, see eLux partitions
- Firmware device configuration of the relevant devices must be configured correctly, see Planning firmware updates

The software delivery can be triggered by a Scout command or a notification.

During delivery, a live information icon is shown on the system bar.

The subsequent installation of the software packages and the update to the new IDF is initiated with an update command or an update notification.

> **Note:**

> If you initiate the subsequent firmware update via an **Update** command with **Format system partition**, the system might have to download additional packages. This is because a **Delivery** command only triggers the download of packages that are not installed on the system partition and are not available on the update partition.

- Performing deliveries via command
- Performing deliveries via notification

Performing deliveries via command

1. Select a device, an OU, a Dynamic Device Group or devices in the **All devices** window.
2. From the context menu, choose **Commands > Delivery...**
3. In the **Command** dialog, specify whether and how long you want to inform the user.
4. To allow users to deny the delivery, select **User can cancel command**.

   On the device, a pop-up dialog is displayed that allows users to cancel or start the delivery.
5. To first clean the update partition of the devices, select the **Clean update partition before delivery** option.

   > **Note:**
   >
   > After cleanup, all files of the current software image must be re-transferred and the dynamic proxy cache rebuilt.

6. Specify a time for the delivery. For further information, see Scheduling commands and Executing commands.
7. Click **Execute**.

The delivery is triggered at the defined time. If there is an updated IDF, and if the required software packages do not exist on the update partition of the relevant devices, the delivery of the software packages is started. The system will only download the packages that are missing. If there is less than 30 MB storage space available on the update partition, old packages are deleted before new packages are transferred.

- To check which files have actually been transferred, view the diagnostic file `/var/lib/rpm/eluxman.log`

In the , for each device, the delivery status is shown in the **Properties** window. During the delivery process, the status is shown, including detailed information about the currently processed action with time stamp. Example:

Delivery in progress (Transfer started -- 2018-08-20 11:34:23) Delivery in progress (Transfer in progress -- 2018-08-20 11:35:45) Delivery in progress (Transfer completed -- 2018-08-20 11:35:48)

For further information, see Command results and update information per device.

Performing deliveries via notification

By using delivery notifications, you send an explicit one-time delivery request to selected devices. The request is evaluated with the next connection of the device to its Scout Server. Then, the delivery of all required software packages for the image configured in the Scout firmware configuration is started for this device.

1. Select a device, an OU, a Dynamic Device Group or devices in the **All devices** window. .
2. From the context menu, choose **Notifications > Initiate software delivery**.

   The **Software delivery notification** dialog is shown.
3. Specify whether and how long you want to inform the user, and if the user is allowed to cancel the command. For further information, see Performing deliveries via command.
4. To clean the update partition before performing the delivery, select the relevant option.
5. Confirm the notification and confirmation.

   The notifications for software deliveries are defined for the relevant devices.

   For each device, in the **Properties** window, the **Delivery notification** field shows the value `Activated`.

   > **Note:**
   >
   > If the **Delivery notification** field in the **Properties** window is hidden, click to define which fields you want to show.

For the relevant devices, a delivery notification is set. As soon as a device restarts and reconnects to its Scout Server, it receives a delivery request and the delivery notification is automatically deleted.

The delivery status of a device is shown in its **Properties** window. For further information, see Command results and update information per device. If an update fails, no efforts will be made to retry.

> **Note:**

> In the Scout Report Generator, you can filter devices by the field **Image delivery notification**.

DELETING THE DELIVERY NOTIFICATION FOR ONE OR MORE DEVICES

Delivery notifications can be deleted before the software is delivered:

- From the context menu, choose **Notifications > Delete delivery notification**.

> **Note:**
>
> Notifications are always set or deleted for all selected devices regardless of whether they are only available for individual devices.

Dynamic proxy

You may use dynamic proxy devices for the software package distribution to all devices of the same subnet. A dynamic proxy is an automatically selected device in a subnet that downloads the relevant software packages from the configured web server, and then provides them to all other devices in the subnet.

The solution is based on the device roles **Provider** and **Consumer**.

The fully automated provisioning (provider) and discovering (consumers) of the proxy service within subnets is realized in eLux® by using the zero-configuration networking implementation **Avahi**.

- Requirements
- Frame conditions and roles
- Update procedure

Requirements

To be able to perform updates by using a dynamic proxy client, next to the eLux® operating system, the following eLux packages must be installed on the devices of the relevant subnet:

- Dynamic Proxy update
- Avahi
- Squid Update Proxy

The proxy client must have an update partition. For further information on update partitions, see eLux partitions.

Frame conditions and roles

The dynamic proxy client concept is based on the following roles:

PROVIDER

REQUIRES
In the device configuration, under **Firmware > Proxy type**, the device must be configured to `Dynamic`.

The provider is the device that acts as the Dynamic Proxy client. All devices with an update partition can be selected for the provider role. Once a provider is selected, the device remains in the provider role for the upcoming updates. In case the provider is not available at the required point in time, another device with an update partition takes over the provider role. The provider is selected automatically and dynamically.

- To exclude devices from the provider role, under **Firmware > Proxy type**, select None.

CONSUMER

REQUIRES
In the device configuration, under [Firmware > Proxy type], the device must be configured to [Dynamic].

All devices of a subnet that are not selected for the provider role are consumers. The consumers perform their update through the provider of the subnet. The consumers do not need to download software packages from the web server.

- To exclude devices from the consumer role, under **Firmware > Proxy type**, select None.

> **Note:**
>
> In the **Firmware** configuration, if `HTTP` is used, the **User** and **Password** fields must remain empty.

Update procedure

UPDATE CHECK

In the case of an update request coming either from the or from the local configuration (), the consumers download the latest IDF file from the web server and check if they need to perform an update.

DISCOVER PROXY SERVICE

If software packages are required, the consumers try to discover the proxy service in the subnet. If there is no provider existing in the subnet so far, one of the devices with update partition automatically takes over the provider role and provides the proxy service.

DOWNLOAD SOFTWARE PACKAGES

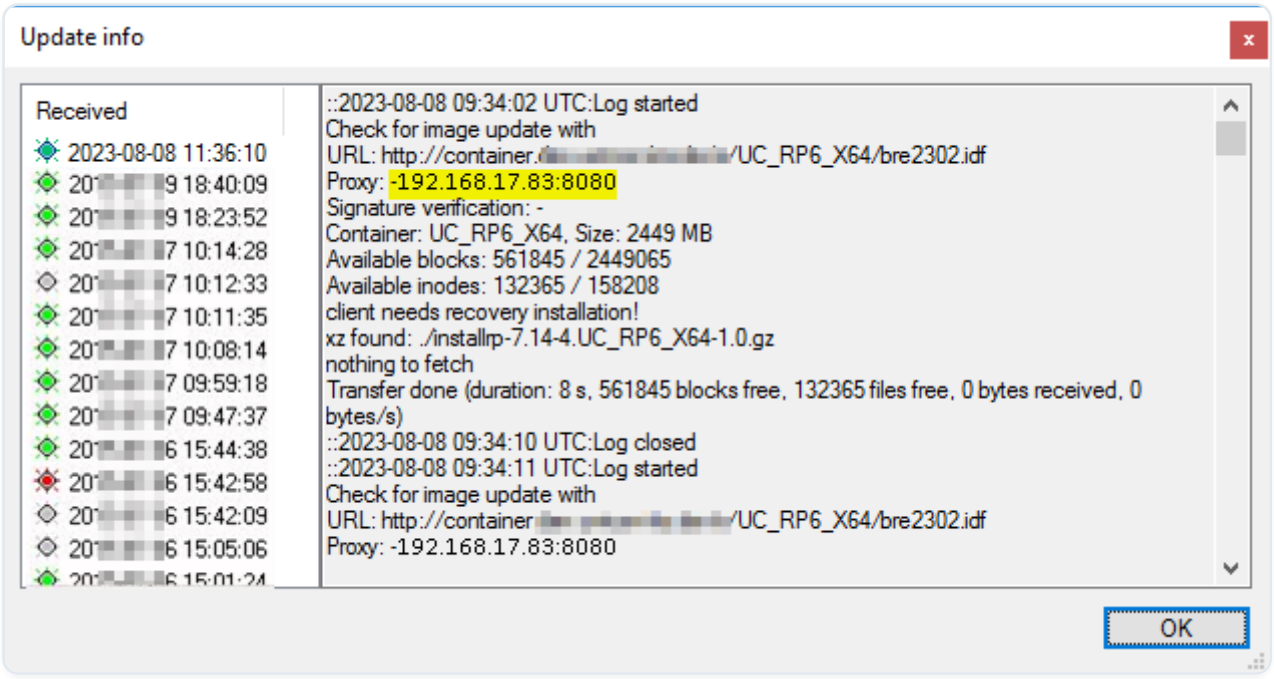
The provider checks the availability of the requested software packages on its update partition and downloads missing packages from the web server specified by the consumers.

DEPLOY AND INSTALL SOFTWARE PACKAGES

The software packages are transferred from the provider to the consumers, and the consumers install the packages. Devices without update partition use the rhythm 'fetch one package - install one package' while devices with update partition fetch all required packages in one step and install them subsequently. Only after the last consumer was provided with the relevant software packages, the provider updates its own system, if needed.

Update information is recorded for both, the consumers and the provider:

- For each updated device, the including the provider can be viewed by double-clickling in the window. For further information, see Command results and update information per device.



- The provider has a local file containing the consumers that have been provided with software packages.

"C:\Users\Sophia.Jose\Documents\bitbucket repos\en-us-unicon\en-us\unicon-elux-scout\media\admin_guides\scout\dia_update_info_proxy.png"

Static proxy

from

If you want to update narrow-band connected devices, you might wish to use a proxy device to forward the firmware update. Proxy devices download the required software packages and distribute them to other devices.

The packages are kept locally in the RAM of the proxy device. Depending on the overall size of the packages defined in the image, 2 GB RAM or more are required.

For the proxy device, **Squid** is used as the proxy server software.

> **Note:**

> Squid supports the `HTTPS` protocol. To make the update process more secure, you may additionally use signatures for the image and software packages. For further information, see Firmware security through signatures.

> The configuration in Scout includes three basic steps:

- Creating an application definition for Squid
- Setting up the proxy device
- Configuring the relevant devices for the proxy update

Creating an application definition for Squid

1. Create a dedicated OU which will be configured for the proxy device.
2. For the new OU, add a local application. On the **Local** tab, edit the following fields:

OPTION	VALUE
Name of application	Squid
Local application	Custom
Parameter	squid
Hidden	On
Start automatically after 0 seconds	On

3. Move the proxy device into the OU and restart it.

The device receives the Squid application definition.

Setting up the proxy device

1. Provide the proxy device with a firmware image including the **Squid Update Proxy** package. To do so, modify the relevant image by using ELIAS, and then update the device.

   > **Important:**
   >
   > The **Dynamic Proxy update** package must not be included.

2. For the OU of the proxy device, open **Device configuration > General** and clear the **Use parent** option.

   Inheritance is disabled and the proxy OU can be configured individually.

3. In **Device configuration > Firmware**, on the **Proxy type** list, select **None**.

4. For the proxy device, open **Device configuration > Network > LAN**, and edit the LAN connection. In the **Edit network profile** dialog, select **Use following IP address**. Leave the **Domain** box empty and confirm with **OK**.

   The last obtained IP address is used as the static IP address by the proxy device.

Configuring devices for the proxy update

1. For the OU or devices you want to receive their firmware updates through the proxy device, open the **Device configuration** dialog.
2. On the **Firmware** tab, edit the following fields:

Protocol	HTTP
Proxy type	Static (Consumer)
Provider	<>:3128
User and Password	<>

3. Edit the further fields as usual, see Configuring firmware updates.

Once the modifications have become active, the relevant devices receive their firmware updates from the proxy device.

Troubleshooting firmware update

Error messages

ERROR MESSAGE	REASON	SOLUTION
Bad container	Containers are hardware-specific.	Check whether the container matches your device specifications.
Bad flash size	IDF size exceeds flash size	Verify whether the image size defined by the IDF matches with the actual flash size of the device.
Bad authorization	Incorrect device password	Correct the entry in **Device configuration > Security**.

ERROR MESSAGE	REASON	SOLUTION
Client needs recovery information	If critical feature packages (.fpm) are updated in the baseOS, the device requires a recovery installation before it can be updated.	For further information see, eLux Recovery procedures in the **eLux Recovery procedures** short guide.

Update options

If the update is still faulty, try to modify the update settings. For further information, see Update/ Delivery

Passwords

eLux® users and administrators may be required to enter passwords in various situations, for example

- AD logon for user authentication after eLux system start
- Application logon
- Device password to unlock the device configuration On the Scout side, the device password applies to all devices of an infrastructure and is used for additional functions.
- Accessing a password-protected OU
- Defining a mirror password
- Setting up network profiles
- Further features of the device configuration Example: Defining a network drive
- To view and verify a password after typing, next to the **Password** entry field, click .
- To reset a password entry, click .

Local device password

The device password affects the local devices. All devices managed by a specific Scout Server receive the same device password so that it is used to verify the access rights for the devices. The device password is requested by Scout for management tasks such as discovery.

The device password can only be changed centrally for all devices in the Scout Console. The initial password is set to elux.

> **Note:**

> We recommend changing the password immediately to avoid unauthorized configuration caused by local users.

> Usually, the access rights do not allow users to modify their local device configuration in the **Security** dialog. However, if the user or administrator changes the device password locally for a device, this device can no longer be managed by Scout.

Changing local device password via Scout Console

Changing local device password on a device

For further information, see Device password in the eLux guide.

Changing local device password via Scout Console

> **Important:**

> Using this function, you change the device password of **all** devices managed by this .

1. In the Scout Console, click **Options > Base device configuration... > Security**. Under **Local security**, click **User rights**.
2. In the **User rights** dialog, in the **Device password** field, type the new password and repeat it in the **Conform device password** field.
3. Confirm with **OK**.

The new device password is assigned to the devices on the next restart.

> **Note:**

> To immediately activate the new password, perform a Scout **Restart** command for the relevant devices. For more details, refer to the commands execute guide.

Changing local device password on a device

1. In the eLux® Configuration panel, click **Security > Device password**.

2. Under **Current password**, type the old password, and in the next two fields type the new password.

   To view your entry after typing, click .

3. Confirm with **Apply**.

> **Important:**
>
> The device can no longer be managed by Scout.

Managing administrators

- Activating administrator policies
- Adding administrators
- Deleting administrators
- Administrator policy
- Viewing administrator activities
- Pass-through authentication
- Maintenance windows

Adding administrators

You may define any AD users or groups as Scout administrators.

> **Note:**
>
> Check whether the relevant SQL Server permissions are set. For further information, see SQL Server users and application roles in the **Installation** guide.

1. In the Scout Console, click **Security > Manage administrators**...
2. In the **Administrator rights** dialog, click **Add Administrators**...

   The **Initial administrator profile** dialog opens.

3. Select the access range for the new admin and confirm with **OK**.

   The Windows **Administrator rights** dialog opens.

4. Below the Group or usernames field, click **Add**...

   The Windows **Select Users or Groups** dialog opens.

5. Enter the relevant AD username or AD group name, and then click **Check Names**.

   Or:

   Search for the AD user or AD group by using the **Advanced**... button.

6. Confirm with **OK..**

The new user or group is added to the list of administrators. You may assign the appropriate object rights to them now. For further information, see Administrator policy.

The administrators can log on to the Scout Console now by using their Windows account information.

> **Note:**
>
> If you use AD groups only, and if a user is a member of more than one group, the access rights of the groups are not consolidated, but the rights of the first group found apply. If users are authorized with their AD users and if they are authorized with one or more AD groups at the same time, the access rights are not consolidated but the rights of the AD user apply.

Deleting administrators

1. In the Scout Console, click Security > Managing administrators.
2. In the Administrator rights dialog, select the relevant administrator.
3. Click Delete administrator.

The selected administrator is deleted from the Scout administrators list without an 'Are you sure?' verification.

Administrator policy

For all Scout administrators there are three different kinds of access rights:

Base rights	Main access rights organized in functional blocks

Menu rights	Access rights for specific menu commands
Object rights	Access rights on OU or device level for properties, device configuration, applications, and some other functions
Default object rights	Default access rights for all OUs or devices for which no different object rights have been defined

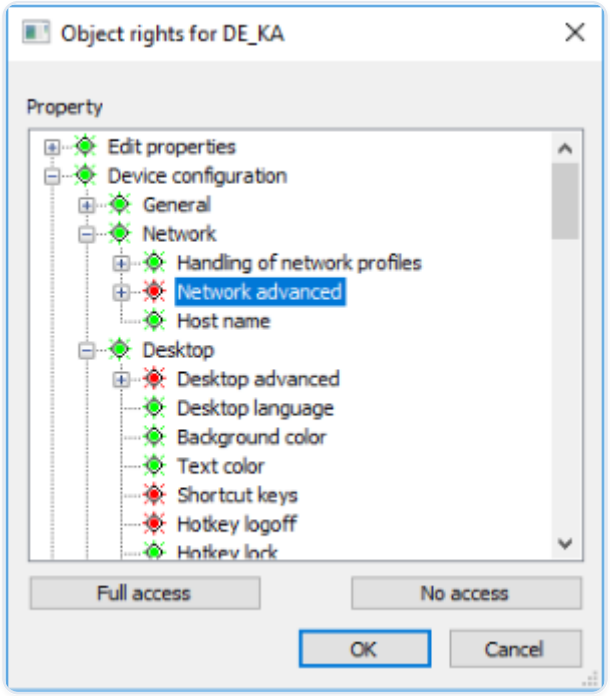
Note that in the dialog you must always first select the relevant administrator for whom you want to edit the access rights. To edit object rights, first select the relevant OU or device.

In the **Administrator rights** dialogs, the provided rights are displayed with a green or red symbol:

Access granted Access denied

To toggle between **Access granted** and **Access denied**, double-click the relevant right or press the space key.

If you click the **Full access** or **No access** buttons, all of the displayed rights are set to green or red, respectively.



Important:

> For all kinds of access rights, the following applies: If a right is disabled, the relevant administrator will no longer have access to the related function. For the last or only administrator existing, you cannot disable access rights. This is to prevent being locked out of the Scout Console

- Changing base rights
- Changing menu rights
- Changing object rights
- Changing default object rights
- Defining a Start OU

> **Note:**

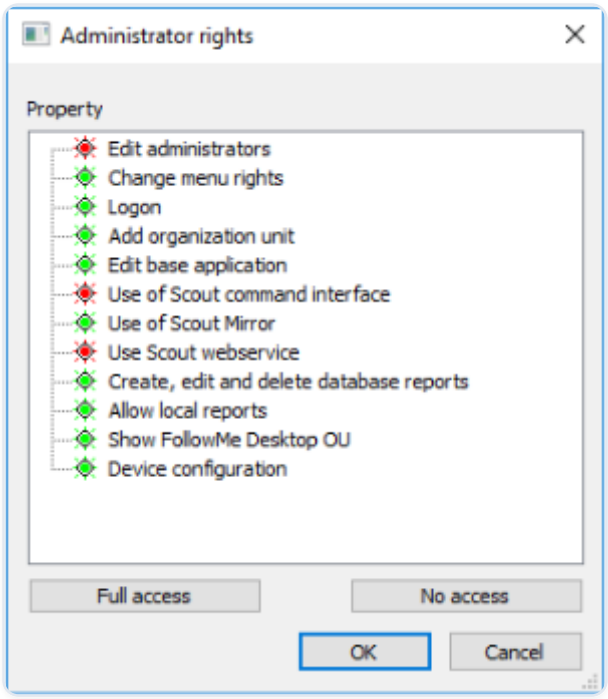
> UniconOS Management Board has its own set of base rights and object rights, some of which only take effect in UniconOS Management Board and are not listed above. For further information, see Administrator rights in UniconOS Management Board.

Changing base rights

Administrator base rights refer to entire functional blocks such as using the Scout Report Generator or configuring FollowMe Desktop.

1. In the Scout Console, click **Security > Managing administrators**.
2. In the **Administrator rights** dialog, select the relevant administrator.
3. Click **Base rights**.

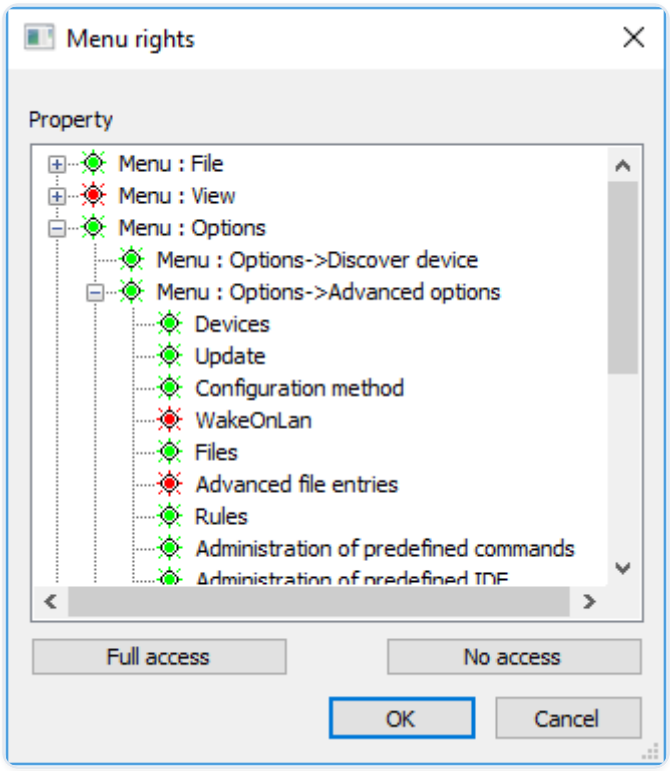
   The **Administrator rights > Base rights** dialog opens.
4. Change the relevant rights by double-clicking or by pressing the space bar.
5. Confirm with **OK**.



Changing menu rights

Menu rights refer to the executability of menu functions. If you deny an administrator access to a menu option, the menu item is dimmed and the administrator cannot perform the function.

1. In the Scout Console, click **Security > Menu rights**.
2. In the **Menu rights** dialog, select the relevant administrator.
3. Click **Menu rights**. The **Menu rights** dialog opens.
4. Change the relevant rights by double-clicking or by pressing the Space bar.
5. Confirm with **OK**.

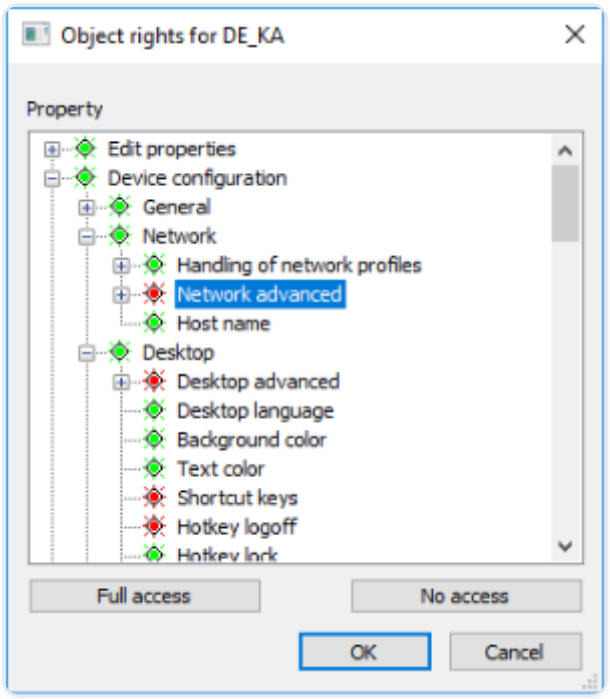


Changing object rights

Object rights refer to an OU or a device. You can define object rights for selected OUs or devices that differ from the general default object rights.

1. In the tree view, select an OU or device.
2. Click **Security > Object rights** or, from the context menu, choose **Object rights**.
3. In the **Object rights** dialog, select the relevant administrator.
4. Click **Edit object rights**.

   The **Object rights for** `<OU>` dialog opens.
5. Change the relevant rights by double-clicking or by pressing the Space bar.
6. Confirm with **OK**.



(/en-us/unicon-elux-scout/2611/media/admin_guides/scout/admin_object_rights.png)

RESETTING OBJECT RIGHTS TO DEFAULT

You can reset object rights that you have defined differently for an OU or a device:

1. In the **Object rights** dialog of the OU or device, select the relevant administrator.
2. Click **Delete object rights**.

Changing default object rights

Default object rights apply to all levels to all OUs or devices for which no specific object rights are defined.

1. In the Scout Console, click **Security > Managing administrators**.
2. In the **Administrator rights** dialog, select the relevant administrator.
3. Click `Default object rights...`.

   The Default object rights dialog opens.
4. Change the relevant rights by double-clicking or by pressing the Space bar.
5. Confirm with OK.

Defining a Start OU

This feature lets you determine that an administrator is allowed to see only a particular start OU including its subordinate OUs.

1. In the Scout Console, click **Security > Manage administrators**.

2. In the **Administrator rights** dialog, select the relevant administrator.
3. Click **Set root OU**.

   The **Root organization unit** dialog opens.
4. Check the **Use the following root organization unit** option.
5. Select the relevant root OU.
6. Confirm with **OK**.

Viewing administrator activities

The activities of all administrators are logged according to the defined monitoring level in **Security > Manage administrators**.

You can view the logs in the Scout Console and you can export them.

1. In the Scout Console, select **Security > Manage administrators**.
2. Under **Monitoring**, click **View protocol**. The Activities dialog opens and shows the activities of all administrators.
3. To export log entries into a text file, select the relevant entries and click **Export selected items**.

Monitoring levels

The monitoring levels 0-3 build on each other. Level 0 only logs changes to the monitoring level itself.[1[from Scout Enterprise Management Suite15 2101]] At each level further activities are added. Logging with monitoring level 3 is the most detailed.

Note:

Use monitoring level 3 only temporarily for diagnostic purposes, since all database transactions are additionally logged.

ACTIVITIES / LEVEL 1	ACTIVITIES / LEVEL 2	ACTIVITIES / LEVEL 3
Log on	Save / Delete / Rename application	All SQL statements executed due to administrator activities in the console

ACTIVITIES / LEVEL 1	ACTIVITIES / LEVEL 2	ACTIVITIES / LEVEL 3
Log off	Upload application	
Deny logon	Save[2[[ ]also applies to Move device]] / Delete / Rename device	
Add administrator	Save[3[[ ]also applies to Move OU]] / Delete / Rename OU	
Edit / Delete administrator rights	Save device configuration	
Change administrator policies	Save advanced device configuration	
Request mirroring of a device	Save advanced options	
Change device password	Discover devices	
	Schedule/change/delete Scout command	
	Set / Delete relocation notification	
	Set / Delete update notification	
	Set / Delete delivery notification	
	Set / Delete UEFI update notification	

Maintenance windows

A maintenance window is a time period that is scheduled for maintenance work. In the , authorized administrators can define maintenance windows to keep this period free for necessary IT

maintenance tasks such as installing server updates. While a maintenance window is active, non-authorized administrators cannot use the and cannot start jobs.

Defining a maintenance window

REQUIRES
Menu right: **Menu: View > System diagnostics > Maintenance windows**

1. In the Scout Console, click the menu entry **View > System diagnostics > Maintenance windows...**
2. Click **Add**.
3. In the **Edit Maintenance window** dialog, specify the following options for your new maintenance window:

OPTION	DESCRIPTION
Name	With descriptive names, you can distinguish multiple maintenance windows.
From	Date and time when the maintenance period will be started for the first time
for	Time period of the maintenance window
	Specify number and unit (hours or minutes)
	Example: 2 hours
Repeat	Repeats the maintenance window periodically, see example below

4. Confirm with **OK**.

The maintenance window is defined and shown in the list of maintenance windows. For the duration of a defined maintenance window, the console will be blocked for administrators who do not have the **Maintenance window** menu right. The affected administrators can then only close the console, other actions are disabled. After the maintenance window has expired, they can restart the console as usual.

> **Note:**

> By default, the menu right is active and maintenance windows have no effect. If you disable the menu right for certain administrators, they will not be able to work with the console during the defined maintenance windows. Administrators with active menu right are not subject to any restrictions.

Example of a periodic maintenance window

For a maintenance window, for example defined from Friday, 06.09.2019, 20:00 for 2 hours, you can set the following repeat options:

Every day	Daily, starting on Friday, 06.09.2019
Every week	Every Friday of a week
Every month	Every 6th of a month
Every first weekday in month	Every Friday of the first week of the month
Every second weekday in month	Every Friday of the second week of the month
Every third weekday in month	Every Friday of the third week of the month
Every fourth weekday in month	Every Friday of the fourth week of the month
Every last weekday in month	Every Friday of the last week of the month

Scout Keep Alive service

The Scout Keep Alive service offers an evaluation of configurable status messages of the managed devices. These status messages are called **keep alive messages**.

Within a defined time interval, the configured devices send status messages to the Scout Keep Alive service. These messages allow to refresh the status of the relevant devices in the Scout Console.

1	The device has no network connection or is switched off.

2	The device is ready for operation. The network connection is operational.

For further information, see Icons in the tree view.

The configuration is part of the device configuration and can be applied to the entire infrastructure or to selected OUs and devices.

The **keep alive messages** are transferred from the devices to the Scout Keep Alive service by using the HTTPS protocol. This requires a valid SSL certificate. For further information, see Certificate for Scout Keep Alive service.

> **Note:**
>
> For Scout versions up to Scout 15 2204, the Scout Statistics Service as an optional component of the Scout Enterprise Management Suite® performed this task. This required an additional statistics database. With Scout 15 2209, the Scout Statistics Service has been replaced by the Scout Keep Alive service: The keep alive data are now stored in the Scout database. The processing of dynamic asset details for statistical analysis is no longer supported.

- Configuring status messages for devices
- Log files of the Scout Keep Alive service

Log files of the Scout Keep Alive service

from Scout 15 2302

For the Scout Keep Alive service, log file rotation is active, which writes information about the service start and errors by default. The log level and other log options may be customized via an `.ini` file.

After installation, the log files and the `scoutkeepalive.ini` are located under `%PUBLIC%DocumentsUniConScoutScoutKeepAlive`

The following settings are available in `scoutkeepalive.ini` in the `Logger` section:

OPTION	DEFAULT	VALUE RANGE	DESCRIPTION
MaxLogFiles	10	1..n	Number of log files for log rotation
MaxLogFileSizeMB	100	100..n	File size in MB
Level	1	1..15	Log level
			1 Service start and errors
			2 Information beyond errors
			4 Program flow
			8 Program flow including further details
			To receive error messages, information beyond errors and information on the program flow, we recommend protocol level 7 (bit mask of 1+2+4).
Location	`C:UsersPublicDocumentsUniconScoutScoutKeepAlive`		Destination for the log files

Import/Export

All import and export functions can be performed via the Scout Console or the SCMD interface.

Exported files are saved in an XML format. The file name extension depends on the data category.

DATA CATEGORY FOR EXPORT/IMPORT	FILENAME EXTENSION
Device configuration of OUs	.oustp
Device configuration of devices	.devstp
Properties of OUs	.oupro
Properties of devices	.devpro
Properties of applications	.apppro
Device list	.csv
OU tree	.outree

- Exporting
- Importing

Exporting

1. Select the OU you want to export data from.
2. Click **File > Export** and what you want to export.
3. Select a folder to save and apply with **OK**.

Importing

You can import device configuration data, device properties and application properties. In addition, you can import device lists and OU trees. The import file must have the relevant file name extension.

1. Select the OU you want to import data into.
2. Click **File > Import** and the data category you want to import.
3. Apply with **OK**.

Log files and optimizing

- Log files
- Optimizing

Log files

Scout provides three logging options which are saved as `.log` files on the Scout Server.

OPTION	LOG FILE	DESCRIPTION
Scout Console	`scout.log`	Required for debugging
		Path: `%USERPROFILE%\Documents\UniCon\ScoutConsole`
		In the Scout Console, click **View > System diagnostics > Console log**.
Scout Server	`eluxd.log`	Log file of the Scout service, required for support calls
		Default path: `%PUBLIC%\Documents\UniCon\ScoutServer`
		Previous versions are renamed in elux.log.1, elux.log.2, etc.
		In the Scout Console, click **View > System diagnostic > Server log** (only if the Scout Console is installed on the same machine as the Scout Server).
Server keep alive log	`keepAlive.log`	Log file for keep alive entries of the Scout Server
		For devices, also a keepalive log file exists in the ScoutKeepAlive directory, provided the Scout Keep Alive service is installed.
		Created every 10 minutes

OPTION	LOG FILE	DESCRIPTION
		Default path: `%PUBLIC%\Documents\UniCon\ScoutServer`
Scout Statistics Service	`UniconStatisticService.log`	Rotating log file with configurable path
		Default path: `%USERPROFILE%\Documents\UniCon\ScoutStatisticService`
		The location and the maximum size from which a new file is written can be configured in `%PROGRAMFILES%\Unicon\ScoutStatistic\statisticservice.exe.config`

For further information about file paths, see Program and file directories.

Note:

Click View > System diagnostic > Server files] to open the server files directory in the Windows File Explorer (if console and server are installed on the same machine). The directory contains all configuration and log files organized in their application directories.

The following additional logs are available via the and can be exported:

OPTION	DESCRIPTION
License log	All actions related to licenses such as entering, deleting and exporting licenses
	View > System diagnostics > License log

OPTION	DESCRIPTION
Administrator activity log	All administrator activities depending on the defined monitoring level
	Security > Manage administrators > View protocol
	For further information, see Viewing administrator activities.

- Enabling logging
- Configuring Scout Server log
- Cleaning up mirror log files

Configuring Scout Server log

For the Scout Server log file `eluxd.log` the `Scout Server` creates more than one backup (Log rotation). Once a new `eluxd.log` is created, the previous version is saved to the file `eluxd.log.1`, while the old `eluxd.log.1` is saved to `eluxd.log.2` and so on.

The log files continue recording when the server is restarted. The creation of a new log file is triggered by the following parameters:

- log file size
- maximum number of log files

You are free to configure both thresholds by yourself.

In addition, you may modify the location of the server log file and the keep alive log file (by default in `%PUBLIC%\DocumentsUniConScoutServer`). Specify any local directory for them, but no network directory.

MODIFYING LOG ROTATION AND LOCATION

1. In the file system, in `%PUBLIC%\DocumentsUniConScoutServer`, open the `eluxd.ini` file for editing.
2. To modify the rotation parameters, add the following entries:

SECTION	ENTRY	DEFAULT	DESCRIPTION
ELUXD	MaxLogFileSizeMB	100	Maximum size of a log file in MB
ELUXD	MaxLogFiles	10	Maximum number of log files (eluxd.log plus backups)

3. To modify the log file location, add the following entry:

SECTION	ENTRY	EXAMPLE	DESCRIPTION
ELUXD	LogFileLocation	c:\log	Local directory to be used for the log files `eluxd.log` and `keepAlive.log`

> **Important:**
>
> Specify a local directory to which the Scout Server can write. Do not use the UNC (Uniform Naming Convention) format.

After the Scout service has restarted, the log files are written to the specified directory.

If the Scout service cannot access the directory, it cannot start and creates an entry in the Windows Event Viewer. If the Scout service is running but cannot write the log file, it creates an alert message in the Scout Console.

Cleaning up mirror log files

-- from Scout 15 2101 --

If configured for mirroring, Scout creates a log file for each mirroring session and stores it in the mirror directory under the files. For further information, see Configuring mirroring .

You can have these log files deleted automatically via a command.

1. In the Scout Console, click **View > System diagnostics > Mirror log cleanup**.

The **Command** dialog opens with the command **Mirror log cleanup**.

2. To select the period relative to the current date, under **Delete log files older than**, enter the number of days before which you want to delete log files.
3. Specify the point in time of the execution. To define periodic scheduling of the cleanup process, select **Every**. Then, specify a day of the month or a weekday and then the time.
4. Confirm with **Schedule**.

Periodic scheduling of the command ensures that mirror log files have only a limited lifetime. Example: If you define **90 days** and **Friday**, the log files will remain available for a maximum of one week longer than 90 days.

> **Note:**
>
> Other log data such as administrator activities, executed commands, or alarms in the Scout Console can be deleted via database cleanup. For further information, see Database cleanup.

Optimizing

To optimize the performance and deal with high network loads you can use the following options:

- Configuring Optimizing with handshake for a device, OU or all devices
- ManagerLoadBalancing to configure load distribution if you use a SQL database For further information, see the `Installation` short guide.
- Configuring the number of ODBC connections if you use a SQL database For further information, see the `Installation` short guide.

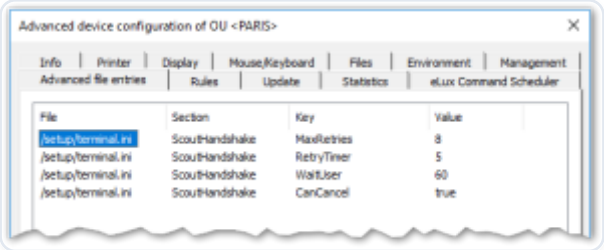
Optimizing with handshake

During each start-up the devices contact their Scout Server and check for new configuration data and application definition data. If they can't access the Scout Server, they retry to connect and synchronize according to their handshake configuration.

Activating new configuration data might require a restart of the client. Then the user is informed and has the chance to suppress restarting.

Handshake parameters can be set in the `terminal.ini` file of the client by using the **Advanced file entries** feature. For further information, see Advanced file entries.

Handshake can be configured for the entire organization or for a particular OU or device.



The values shown in the figure above are examples and can be modified. By default, handshake is not configured.

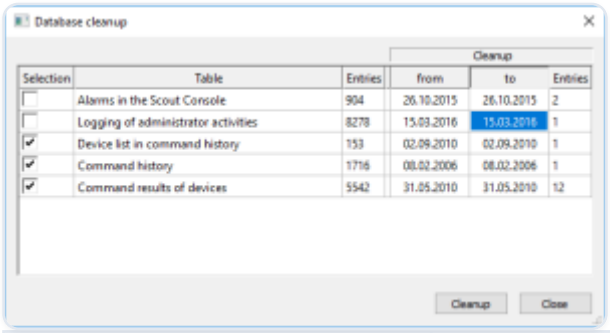
The `ScoutHandshake` section provides the following configurable parameters:

PARAMETER	DESCRIPTION
MaxRetries	Number of connection attempts. The value 0 deactivates handshake.
RetryTimer	Period of time in seconds until next connection attempt (start value). After each attempt, the interval is doubled (+/- random value). Example: Having defined 8 connection retries and a **RetryTimer** start value of 5 seconds, the 8th connection attempt is carried out after about 21 minutes.
PermanentRetriesAfterDays	Number of days (maximum) from the last successful connection until next connection attempt. Ensures that after n days latest the configuration data of device and [Scout Server] is compared. Can be combined with **MaxRetries** and **RetryTimer**.
WaitUser	Waiting time before client restarts to give the user the chance to close applications or log off.
CanCancel	Defines if the user is allowed to suppress a client restart (true\false).

Database cleanup

Scout stores huge amounts of data concerning various processes such as any performed update commands. To purge the Scout database tables, authorized administrators can delete database entries from particular tables for a specified period of time.

The relevant tables are listed in the **Database cleanup** dialog, each table providing the total number of entries and the creation date of the first entry. The administrator can modify only the fields **Selection** and **to**.



Alerts in the [Scout Console]	Alert messages (Error, Warning, Info), can be viewed by double-clicking the lamp icon on the Scout Console status bar	☐
Logging of administrator activities	Log file entries about the activities performed by an administrator according to the defined monitor level in **Security > Manage administrators** For further information, see Viewing administrator activities.	☐
Device list in command history	History data of commands shown in **View > Command history** of the Scout Console (entries for particular devices)	
Command history	History data of commands shown in **View > Command history** of the Scout Console (entries for OUs)	
Command results of devices	Results of commands performed on the devices (Update, Delivery, user-defined command). The log	

	data can be viewed in the Scout Console in the **Properties** window or by using the context menu of a device **Commands > Update/Delivery/Command**.	
	For further information, see Command results and update information per device.	

PERFORMING A DATABASE CLEANUP

1. Select **View > System diagnostics > Database cleanup**.
2. In the **Database cleanup** dialog, for the relevant table, in the "to" field, specify a date that indicates the end of a time span for the deletion of the entries (all entries up to and including this date are deleted).

   The **Entries** column at the right shows the number of entries to be deleted.
3. Click into the field of the `Selection` column on the left to activate this table for cleanup.

   A check mark indicates that entries from this table are selected for cleanup.
4. Click **Cleanup**.

   A message shows the total of all entries in all tables which are intended for cleanup.
5. Confirm with **Yes**.

From the selected tables, all entries up to the specified dates are deleted.

> **Note:**
>
> Before you can delete command history entries, you are required to delete the corresponding device list entries.

UniconOS Management Cloud

Overview

With UniconOS Management Cloud, Citrix® offers a fully managed cloud infrastructure for the complete UniconOS endpoint management solution. The infrastructure is setup automatically, and the customer administrators get access to the endpoint management applications and can focus on the endpoint management.

With UniconOS (elux), Citrix provides a lean and secure operating system which runs on endpoints like thin clients, desktop PCs, and laptops.

With UniconOS Management (Scout) the provided components enable administrators to manage all endpoints from a central location.

The management components consist of Scout, Elias and Scout Cloud Gateway (SCG).

Scout is the central management component running on a Windows Server. Scout consists of multiple components, each running as a service: Scout, UniconOS Management Board, DBLayer, Keep Alive Service.

Elias enables administrators to create and manage software images and packages. An image makes up the complete UniconOS (eLux®) system (operating system, drivers and applications), which runs on the endpoint.

Scout Cloud Gateway (SCG) is a VPN Gateway solution, which gives the possibility to securely connect UniconOS endpoints with UniconOS Management Cloud, so that these endpoints can be managed.

Running and maintaining the management components requires to setup the infrastructure, network and all the components. The required network and firewall rules need to be set and maintained. Components like SCG need to have a scalable infrastructure and high availability. Elias needs to be able to provide the images and packages to the endpoints and needs to be accessible.

When running on premises, the management components need to be set-up, maintained, updated, and managed by the customers' IT departments, while the endpoint administration will be performed by the same or dedicated administrators. This takes time and effort, and it is costly.

Nowadays, cloud offerings are common practice and software as a service is commonly used by many companies. The infrastructure is managed by the provider, and complex requirements like

high availability, scalability, UpToDate software, backup and disaster recovery and support are part of the offering.

When using UniconOS Management Cloud, instead of managing the whole infrastructure on premises, you can work with a ready to use cloud environment and focus on the endpoints and the end user experience. While the endpoint management solution is part of the cloud environment, administrators can use the web-based application for performing endpoint management tasks.

Endpoint Management Components

In a UniconOS Management Cloud environment, the following components exist:

- Scout (UniconOS Management)
  - Scout Server
  - UniconOS Management Board
  - Scout Board DBLayer
  - Scout KeepAlive
- Elias 18 (Image and package management)
  - Elias Server
- Scout Cloud Gateway (SCG ~ Remote endpoint management)
  - SCG Services
  - OpenVPN Server

To enroll remote devices with tokens and manage their certificates, see Enrollment settings.

Architecture

The UniconOS Management Cloud production environment facilitates the required resources and provides the optimal environment for production. The following features are key to the production environment.

- High availability
- Scalability
- Backup and disaster recovery (Full backups are being performed daily)
- Centralized logging (Logs are aggregated centrally for easy troubleshooting)
- Monitoring

- 24/7 Support
- Software updates and maintenance
- Image and package distribution via Blob Storage
- Multi-region

The database is based on SQL for endpoint management (Scout). For Elias and SCG, Document DB is being used.

The administrator can access the management applications using a web browser, and the communication is secured via HTTPS.

The communication between the endpoint and the management component is secured via HTTPS (Elias software packages) and TLS (bidirectional communication between eLux and Scout and eLux and SCG).

> **Info**
>
> Please contact your ATS for the next steps.

UniconOS Management Cloud

Overview

With UniconOS Management Cloud, Citrix® offers a fully managed cloud infrastructure for the complete UniconOS endpoint management solution. The infrastructure is setup automatically, and the customer administrators get access to the endpoint management applications and can focus on the endpoint management.

With UniconOS (elux), Citrix provides a lean and secure operating system which runs on endpoints like thin clients, desktop PCs, and laptops.

With UniconOS Management (Scout) the provided components enable administrators to manage all endpoints from a central location.

The management components consist of Scout, Elias and Scout Cloud Gateway (SCG).

Scout is the central management component running on a Windows Server. Scout consists of multiple components, each running as a service: Scout, UniconOS Management Board, DBLayer, Keep Alive Service.

Elias enables administrators to create and manage software images and packages. An image makes up the complete UniconOS (eLux®) system (operating system, drivers and applications), which runs on the endpoint.

Scout Cloud Gateway (SCG) is a VPN Gateway solution, which gives the possibility to securely connect UniconOS endpoints with UniconOS Management Cloud, so that these endpoints can be managed.

Running and maintaining the management components requires to setup the infrastructure, network and all the components. The required network and firewall rules need to be set and maintained. Components like SCG need to have a scalable infrastructure and high availability. Elias needs to be able to provide the images and packages to the endpoints and needs to be accessible.

When running on premises, the management components need to be set-up, maintained, updated, and managed by the customers' IT departments, while the endpoint administration will be performed by the same or dedicated administrators. This takes time and effort, and it is costly.

Nowadays, cloud offerings are common practice and software as a service is commonly used by many companies. The infrastructure is managed by the provider, and complex requirements like high availability, scalability, UpToDate software, backup and disaster recovery and support are part of the offering.

When using UniconOS Management Cloud, instead of managing the whole infrastructure on premises, you can work with a ready to use cloud environment and focus on the endpoints and the end user experience. While the endpoint management solution is part of the cloud environment, administrators can use the web-based application for performing endpoint management tasks.

Endpoint Management Components

In a UniconOS Management Cloud environment, the following components exist:

- Scout (UniconOS Management)
    - Scout Server
    - UniconOS Management Board
    - Scout Board DBLayer

    - Scout KeepAlive
- Elias 18 (Image and package management)
    - Elias Server
- Scout Cloud Gateway (SCG ~ Remote endpoint management)
    - SCG Services
    - OpenVPN Server

To enroll remote devices with tokens and manage their certificates, see Enrollment settings.

Architecture

The UniconOS Management Cloud production environment facilitates the required resources and provides the optimal environment for production. The following features are key to the production environment.

- High availability
- Scalability
- Backup and disaster recovery (Full backups are being performed daily)
- Centralized logging (Logs are aggregated centrally for easy troubleshooting)
- Monitoring
- 24/7 Support
- Software updates and maintenance
- Image and package distribution via Blob Storage
- Multi-region

The database is based on SQL for endpoint management (Scout). For Elias and SCG, Document DB is being used.

The administrator can access the management applications using a web browser, and the communication is secured via HTTPS.

The communication between the endpoint and the management component is secured via HTTPS (Elias software packages) and TLS (bidirectional communication between eLux and Scout and eLux and SCG).

> **Info**
>
> Please contact your ATS for the next steps.

Enrollment settings

Enrollment settings is where you manage device-enrollment tokens and certificates for enrolling remote UniconOS devices in UniconOS Management Cloud.

> **Note:**
>
> This is available in **UniconOS Management Cloud** only. On-premises installations don't show this page and continue using Scout Cloud Gateway.

> Go to **Preferences** > **Advanced options** > **Enrollment settings**. The page has two sections: **Certificates** and **Token management**.

Administrators need the **Enrollment settings** administrator right to see this page and its related certificate columns and revoke actions elsewhere in UniconOS Management Board. This right isn't restricted by OU: an administrator with the right can run **Revoke all certificates** on every enrolled device in the tenant, including devices in OUs they otherwise can't see.

To grant the right, go to **Security & permissions** > **Manage administrators**, select the administrator, and click **Edit**. Then select **Enrollment settings** under **Menu rights** > **Menu : Options** > **Menu : Options -> Advanced options**, and click **Apply**.

A new administrator created with **Minimum access** doesn't get this right. After an upgrade, an existing administrator keeps it until you remove it — except an administrator who was created with minimum access in an earlier version and whose rights were never edited since.

Certificates

Set how long a device's certificate is valid, and, if your tenant has automatic certificate renewal on, how soon before expiry a device renews it:

SETTING	DESCRIPTION
Certificate validity	The certificate's lifetime in days, from 1 to 90.

SETTING	DESCRIPTION
Renewal threshold	How many days before expiry a device renews its certificate. At least 1 day, less than the validity, and at most 60 days. Appears only when automatic certificate renewal is on for your tenant.

Click **Apply settings** after changing a value. UniconOS Management Board rejects values outside these ranges.

> **Note:**
>
> Automatic certificate renewal itself can't be turned on or off in UniconOS Management Board. While it's off for your tenant, device certificates are not renewed, and a device can no longer connect once its certificate expires.

> **Revoke all certificates** revokes the certificate of every enrolled device in the tenant, after you confirm. Affected devices can't connect again until they're enrolled once more.

Token management

A device enrolls using a token that you create here. Enrollment accepts tokens only — other authentication methods, such as Active Directory or an identity provider, aren't available for enrollment.

Tokens are listed under two tabs:

TAB	COLUMNS	ROW MENU
Active	Token (with its description), Organizational unit, Validity ("Expires" date), Token type (Persistent or One-time)	Edit, Deactivate
Deactivated	Token, Organizational unit, Deactivated on, Deactivation reason (Used (one-time), Expired, Deactivated)	Delete icon removes the record permanently

Search tokens filters by the token or its description.

Adding a token

1. Click **Add token**.
2. Fill in the new token's details:

FIELD	DESCRIPTION
Token	The value a device uses to enroll. See the token rules below. Cannot be changed later.
Description	Optional, up to 255 characters. Letters A-Z and a-z only (no umlauts or accented letters), numbers, spaces, and hyphens.
Organizational unit	Required.
Validity	How long the token stays valid: 1 hour, 24 hours (default), 7 days, 30 days, 90 days, or 365 days.
One-time token	Off creates a persistent token, reusable until it expires or you deactivate it. On deactivates the token automatically after its first use.

3. Save the token.

A token must follow these rules:

- It has 12 to 50 characters.
- It contains only letters (A-Z and a-z), numbers, and these symbols: `@ # . _ -`.
- It has at least 5 different characters.
- It is unique in the tenant.

A tenant can have up to 350 active tokens. When the limit is reached, **Add token** is disabled. Expired, used, and deactivated tokens don't count, so deactivate tokens that you no longer need.

Editing or deactivating a token

Use the row menu of an active token to **Edit** its description, organizational unit, or validity (**Keep current** by default), or to **Deactivate** it. You can't change a one-time token to a persistent token.

A deactivated token can no longer enroll a device. Tokens also move to **Deactivated** automatically once they expire, or once a one-time token is used.

Device list

Enrolled devices show three extra columns in **Devices** > **OU structure** (not in **All devices**):

COLUMN	SHOWS
Certificate expires at	The device's certificate expiry date.
Certificate status	**Active**, or **Not active** if the certificate is revoked or expired.
eLux® version	The eLux version the device had when it last enrolled or re-enrolled. Doesn't update with later firmware updates.

These columns are empty for devices that aren't enrolled, and can't be sorted or filtered.

Enrollment > **Revoke certificate**, in a device's context menu, appears only while its certificate is active. For several selected devices, use **Selection** > **Revoke certificates** — it revokes the active certificates in the selection, skips devices without one, and reports the result as "Revoked N; skipped K without a certificate."

A device whose certificate is revoked can no longer connect to Scout.

Enrolled devices elsewhere in UniconOS Management Board

For an enrolled device, UniconOS Management Board hides the **Peer IP address**, **Connected via SCG**, and **SCG address** fields that still show for devices connected through Scout Cloud Gateway.

UniconOS Management Board also locks seven `/setup/terminal.ini` keys that the enrollment process writes on the device (`[Network]` ScoutWebSocketServiceUrl, ScoutCloudEnabled, DesUrl, DeviceCertificatePath, DeviceKeyPath, DeviceCertificateValidUntil, and `[Global]` ClientIdentifier). You can't add or change these keys, in an advanced device configuration, for an enrolled device or for an OU or base configuration that contains one. You can delete an existing entry for one of these keys — if the same entry also exists on a parent OU or the base configuration, delete it there too, or the inherited value keeps applying.

Request diagnostic files works for enrolled devices. Diagnostic files that a user sends from the device itself aren't accepted.

Known limitations

- **Revoke all certificates** can fail for a tenant with more than 100 enrolled devices — the request is sent in a single batch with a 100-device limit.
- **Token management** lists only the 25 most recently created tokens, with no paging. An older active token outside that list can't be seen or deactivated here.
- UniconOS Management Board doesn't check a description against the rules above before sending it. A description with other characters is rejected by the service with the generic error "Request failed with status code 400."

UniconOS Management Board (Scout Board)

UniconOS Management Board is a modern and intuitive management solution for eLux® end points. The web-based interface allows you convenient secure access to your Scout infrastructure. It manages UniconOS devices as well as Windows-based devices that have installed the Scout Agent for Windows. From UniconOS Management 2611, UniconOS Management Board is the only management interface: the Scout Console is end of life.

UniconOS Management Board in UniconOS Management 2611 uses the Citrix® colors, fonts and logos, and it shows the product name UniconOS Management, for example in the browser tab. For details, see Interface.

You continue to use your existing SQL Server database for Scout. Your entire device infrastructure with device configuration, defined applications and access rights structure can now be used via the new UniconOS Management Board interface.

> **Note:**
>
> **UniconOS Management Board** supports Microsoft SQL Server and SQL Server Express, but not SQL LocalDB.

> As part of the Scout Enterprise Management Suite® installation, the UniconOS Management Board is published for the first time with the software release of Scout 15 2209.

The functionality realized so far maps common use cases:

- A dashboard, which gives the administrator a quick overview of some important KPI's
- Access to devices via OU (organization unit) structure with organization units

- Access All devices
- Apply filters to device lists, sort the data, organize data columns
- Dynamic Device Groups (View only)
- Create custom views to manage dynamic device lists
- Search for navigation elements, OUs and devices
- View details for selected devices, OUs and schedules
- Schedule and execute commands on OUs and devices
- Mirror eLux sessions
- Rename, move, delete devices
- Download device diagnostics files
- View and edit device configurations and advanced device configurations for devices and OUs
- Manage settings and preferences
- Schedule device and OU actions
- View and edit device configurations and advanced device configurations for devices and OUs
- View and edit base and base advanced device configurations
- Manage applications
- Manage administrators, permissions and object rights
- Manage the command history
- Compare device configurations
- FollowMe Desktop
- Database cleanup
- Dynamic device configuration (from Scout version 16 2601)
- Image management in **Software**: containers, software packages, images and certificates (from UniconOS Management 2611)
- A REST-API for automating use-cases or integrating eLux management features in custom applications

Note:

The REST API and its reference are part of UniconOS Management Board. For how to open the reference and how scripts sign in, see Using the REST API.

This already provides administrators with a simple and intuitive interface to monitor OUs and devices and provide support to end users. The functionality will be continuously extended and expanded with the following versions.

Note:

Logon via Active Directory is supported.

Overview

UniconOS Management Board is a modern and intuitive management solution for eLux® end points. The web-based interface allows you convenient secure access to your Scout infrastructure.

From UniconOS Management 2611, UniconOS Management Board is the only management interface: the Scout Console is end of life.

UniconOS Management Board in UniconOS Management 2611 uses the Citrix® colors, fonts and logos, and it shows the product name UniconOS Management, for example in the browser tab. For details, see Interface.

You continue to use your existing SQL Server database for Scout. Your entire device infrastructure with device configuration, defined applications and access rights structure can now be used via the new UniconOS Management Board interface.

Note:

UniconOS Management Board supports Microsoft SQL Server and SQL Server Express, but not SQL LocalDB.

As part of the Scout Enterprise Management Suite® installation, the UniconOS Management Board is published for the first time with the software release of Scout 15 2209.

The functionality realized so far maps common use cases:

- A dashboard, which gives the administrator a quick overview of some important KPI's
- Access to devices via OU (organization unit) structure with organization units

- Access All devices
- Apply filters to device lists, sort the data, organize data columns
- Dynamic Device Groups (View only)
- Create custom views to manage dynamic device lists
- Search for navigation elements, OUs and devices
- View details for selected devices, OUs and schedules
- Schedule and execute commands on OUs and devices
- Mirror eLux sessions
- Rename, move, delete devices
- Download device diagnostics files
- View and edit device configurations and advanced device configurations for devices and OUs
- Manage settings and preferences
- Schedule device and OU actions
- View and edit device configurations and advanced device configurations for devices and OUs
- View and edit base and base advanced device configurations
- Manage applications
- Manage administrators, permissions and object rights
- Manage the command history
- Compare device configurations
- FollowMe Desktop
- Database cleanup
- Image management in **Software**: containers, software packages, images and certificates (from UniconOS Management 2611)
- A REST-API for automating use-cases or integrating eLux management features in custom applications

> **Note:**
>
> The REST API and its reference are part of UniconOS Management Board. For how to open the reference and how scripts sign in, see Using the REST API.

> This already provides administrators with a simple and intuitive interface to monitor OUs and devices and provide support to end users. The functionality will be continuously extended and expanded with the following versions.

> **Note:**

> Logon via Active Directory is supported.

Installing UniconOS Management Board

> **Note:**

> UniconOS Management Board is installed as a component of the Scout Enterprise Management Suite®. From UniconOS Management 2611, UniconOS Management Board is the only management interface.

1. Download the latest version of the Scout Enterprise Management Suite and start the installation process. For further information, see Installing Scout Enterprise Management Suite.
2. Fresh installation: By default, the UniconOS Management Board is part of the installation. Upgrading a previous version without UniconOS Management Board: Activate the UniconOS Management Board component.

3. Then configure your UniconOS Management Board settings.

	INFORMATION
1	Number of the port on which the Scout Board service should run
2	Computer name (FQDN) of the machine on which the database layer is to run

	INFORMATION
3	With **HTTPS**, a secure connection to the interface is used.
	Initially, a self-signed certificate is created. To configure an individual SSL certificate instead, see below.

1. Follow the further instructions.

Note:

From 15.2609, an update to UniconOS Management Board keeps hand-made changes to the `.env` file. Before 15.2609, every update rewrote `.env` using only the installer's own values, so any manual change — for example the settings described in the following sections — had to be reapplied after each update.

Every update still rewrites `USEHTTPS`, `DBLAYER_TIMEOUT` and `ZERO_MQ_IS_CURVE_ACTIVATED`, regardless of your own values. The settings of a sign-in method you aren't using are also removed. Comment lines in the file are not kept. Setup never changes the file `.env.custom`, which UniconOS Management Board reads after `.env`.

From 15.2611, a repair of the installation keeps the `.env` file as it is.

From 15.2611, an update also keeps your list of sign-in domains in `DOMAINS`, see Configuring AD domains for UniconOS Management Board.

From 15.2611, an update keeps your own value of `JWT_EXPIRATION`, except `4h`, the value that earlier versions wrote, which it changes to `30m`. `JWT_EXPIRATION` is now the time without activity after which a UniconOS Management Board session ends, see Configuring session limits for UniconOS Management Board.

Installing Software Management

Software Management is the image management in UniconOS Management Board, see Software. The installer has two components for it. A new installation selects both by default, also an unattended one:

COMPONENT	WHAT IT INSTALLS
Software Management	The **Software** section of UniconOS Management Board. You can select it only together with UniconOS Management Board.
Software Management backend	The Windows service **Software Management backend**, which stores the containers and builds the images

An update from an earlier version adds the two components only if you select them. To add them later, run setup again, choose **Modify**, and select both components, see Installing additional features or uninstalling unneeded features.

On the page **Please configure the Software Management backend**, you set the port of the backend (default `22170`) and the **Directory for the Software Management backend data** (default `C:\ProgramData\Unicon\Software`). The containers are in its `containers` subfolder, and the log files of the backend in its `logs` subfolder. Don't use the port of another Scout component, for example `22150` of the Database Layer. Setup then shows "This port is already used by another Scout component. The backend will not start on it."

The data grows with every UniconOS version that you import, to 20 GB and more. Choose a drive with enough space, ideally a disk or partition of its own, so that a full system drive can't affect the operating system. Setup warns if the drive has less than 20 GB free. It refuses a drive root, a folder with a link or junction on the way, a folder that doesn't belong to Administrators, and a folder that already holds other data. Make sure that no unauthorized person can change the data in this folder.

Back up the data folder together with the Scout database, because the database holds the list of your containers, packages and images.

UniconOS Management Board reaches the backend at the address in `ELIAS_TARGET` in its `.env` file. To change this address, run setup again, and enter the new address in **Software Management backend URL** on the **Scout Board** page. The backend always uses HTTPS. Next to UniconOS Management Board, setup gives it a self-signed certificate that UniconOS

Management Board trusts. This certificate is valid for 10 years, and an upgrade in its last year renews it. A backend next to UniconOS Management Board accepts connections only from its own computer, so setup opens no firewall port for it.

So that the **Firmware settings** of an image show the full address that devices download from, set `DEVICE_DOWNLOAD_URL` in the file `config\.env` of the backend, see Configuring the devices.

If administrators reach UniconOS Management Board through reverse proxies, Software Management records the address of the last proxy instead of the address of the administrator. To record the address of the administrator, set `ELIAS_PROXY_TRUSTED_HOPS` in the file `.env.custom` of UniconOS Management Board to the number of proxies that add an address to the header `X-Forwarded-For`, and restart UniconOS Management Board. The default is `0`. Don't set a higher number than you have proxies, because a browser could then choose the address that Software Management records.

Software Management backend on another computer

The installer can also install only the **Software Management backend**, on a computer of its own. Install UniconOS Management Board first, because the backend needs the public keys of UniconOS Management Board and the certificate authority (CA) of its client certificate.

1. When you install UniconOS Management Board, select **Software Management**, clear **Software Management backend**, and enter the address of the backend in **Software Management backend URL**, in the form `https://<host>:<port>`.

   Setup accepts only this form. For an `http://` address or an address with a path, **Next** stays disabled, and setup shows "Use https://host:port for a backend on another machine, without a path or credentials."

   UniconOS Management Board creates its client certificate authority (CA) when it starts for the first time. Then copy the file `elias-client\ca.crt` from the Scout Board folder to the backend computer; the last setup page names the file. The backend setup asks for it. If the backend is installed already, add the content of `ca.crt` to `config\tls\client-ca.pem` in the program folder of the backend instead, and restart the service **Software Management backend**.
2. On the backend computer, start the installer and select only **Software Management backend**.
3. On the page **Please connect the backend to Scout**, enter the name or IPv4 address of the Scout server and the port of its Database Layer, by default `22150`, under **Scout Server DBLayer**, and the **DBLayer server key**. Under **UniconOS Management Board keys**, enter

Session and **Grant**. Under **UniconOS Management Board client CA**, select your copy of `ca.crt`.

On the UniconOS Management Board computer, the DBLayer server key is `ZERO_MQ_PUBLIC_KEY` in `<install folder>\scoutboard\.env`. The public keys of UniconOS Management Board are the registry values `SessionPublicKeys` and `GrantPublicKeys` under `HKLM\SOFTWARE\Unicon\Scout\ScoutBoard`. To show the registry values, run `reg query HKLM\SOFTWARE\Unicon\Scout\ScoutBoard`. Copy the registry values, not a key from the `.env` file.

4. Under **Certificate**, choose **Self-signed**, **PEM files** or **PFX file**.
5. On the **Firewall** page, choose **Automatically**, so that setup opens the port of the backend in the Windows Firewall.
6. On the Scout Server, open `dblayer.ini` in an editor that runs as administrator. Add the IPv4 address of the backend computer to `DBLayerAllowedClients`, see Configuring the Database Layer client check.
7. Then register the backend on the Scout Server: in a command prompt that runs as administrator, run the command line that the last setup page on the backend computer shows, for example `ScoutInstaller.exe /s /v"DBLAYERELIASKEYS=<key>"`. Setup adds the key to `DBLayerEliasKeys` in `dblayer.ini` and restarts the Database Layer, which then also reads the new list of allowed hosts.
8. With a self-signed certificate, copy `config\tls\certificate.pem` of the backend to the UniconOS Management Board computer. Then name the copy in `ELIAS_CA_PATH` in `scoutboard\.env.custom`, and restart the **Scout Board** service.

   System Status in UniconOS Management Board then shows **The ELIAS backend is responding normally.** Until UniconOS Management Board trusts the certificate of the backend, it shows **The ELIAS backend cannot be reached.**

UniconOS Management Board uses a backend on another computer only over HTTPS. If `ELIAS_TARGET` holds an `http://` address, for example after a change by hand, **Software** isn't available, and **System Status** shows **No valid ELIAS backend address is configured.**

The clocks of the two computers must not differ by more than one minute.

An upgrade in the last year of the self-signed certificate renews it. After this upgrade, copy the new `config\tls\certificate.pem` to the UniconOS Management Board computer again, over the file that `ELIAS_CA_PATH` names. UniconOS Management Board takes over the new certificate without a restart.

Replacing the certificate of the backend

1. Copy the new certificate and its private key, both in PEM format, over `certificate.pem` and `private-key.pem` in the `config\tls` folder of the backend's program folder. By default, the program folder is `C:\Program Files\Unicon\Scout\SoftwareManagement`. To find the program folder, run `(Get-CimInstance Win32_Service -Filter "Name='ScoutSoftwareManagement'").PathName` in PowerShell.

   The certificate file starts with the certificate, followed by its intermediates. The key has no password. To use a PFX file instead, run the installer again with this file.

2. Restart the service **Software Management backend** (`ScoutSoftwareManagement`).

3. On the UniconOS Management Board computer, check `ELIAS_CA_PATH` in `scoutboard\.env.custom`. If it names the old certificate, copy the new certificate over this file. UniconOS Management Board takes it over without a restart. For a certificate from a CA that this computer trusts, remove the line instead, and then restart UniconOS Management Board.

Don't set the certificate in the file `config\.env` of the backend yourself. Setup manages the settings `TLS_CERT_FILE` and `TLS_KEY_FILE` there and resets changes to them at the next update or repair. If you add `TLS_PFX_FILE` next to them, the backend doesn't start.

Uninstalling the backend

An uninstallation removes the backend, but it keeps the data folder with the containers and the log files. The Scout database keeps the list of your containers, packages and images. If you install the backend again on the same Scout database, your containers are back.

The uninstallation also removes the secret that protects the private keys of your signing certificates. After a new installation, signing an image then fails with the message "Failed to decrypt signing key". To keep your signing certificates usable, note the registry value `CertificateKeySecret` under `HKLM\SOFTWARE\Unicon\ScoutEliasBackend\Secrets` before you uninstall. Then install again unattended, and add `/v"ELIAS_CERTIFICATE_KEY_SECRET=<value>"` to the command line, see Unattended installation. Keep the value as safe as the private keys themselves.

Installing again on a new Scout database

Containers that are already in the data folder show up in UniconOS Management Board only after the backend has started once with `FORCE_RESCAN=true` in its `config\.env` file. Set it back to `false` afterwards, because a full rescan takes long.

Configuring an individual HTTPS certificate for UniconOS Management Board

Self-signed certificates are not accepted as secure by the browser. To provide the browser with a valid SSL certificate for UniconOS Management Board perform the following steps:

1. Rename your certificate file to `certificate.crt` and the private key file to `private.key`
2. Copy both files into the UniconOS Management Board program directory. By default, this is `C:\Program Files\Unicon\Scout\Scout Board`
3. Still in the UniconOS Management Board program directory, edit the `.env` file and change the entry `CERTIFICATE=SELFSIGN` to `CERTIFICATE=SIGN`
4. Restart the Scout Board service.

For further information and detailed certificate configuration, see HTTPS Certificate Setup.

Configuring authentication options for UniconOS Management Board

During installation, you can choose which authentication method should be used for UniconOS Management Board.

The following authentication methods are supported:

- AD (Active Directory)
- Entra ID (SAML or OIDC)
- OKTA (SAML or OIDC)

With AD, users sign in on the logon screen of UniconOS Management Board, see Logging in. With Entra ID or Okta, UniconOS Management Board opens the sign-in page of the identity provider instead.

After an authentication method has been configured during the installation, you can change or reconfigure it in the `.env` file. As a reference and an example, the template file `.env.example` can be used.

AUTHENTICATION METHOD	PARAMETER	VALUE
AD	`AUTH_TYPE`	`AD`
OIDC	`AUTH_TYPE`	`OKTA_OIDC` or `ENTRAID_OIDC`
	`OIDC_ISSUER`	`your-issuer-url`

AUTHENTICATION METHOD	PARAMETER	VALUE
	`OIDC_CLIENT_ID`	`your-client-id`
	`OIDC_CLIENT_SECRET`	`your-client-secret`
SAML	`AUTH_TYPE`	`OKTA_SAML` or `ENTRAID_SAML`
	`SAML_SSO_ENTRY`	`your-sso-entry-url`
	`SAML_LOGOUT_URL`	`your-logout-url`
	`SAML_SERVICE_PROVIDER_ISSUER`	`https://scoutboard.com/47d899fb-3002-4692-b741-8068b8f7fbb4` Only replace, if you are not using the UniconOS Management Board default issuer
	`SAML_CERT_PATH`	`your-certificate-path`
	`SAML_PRIVATE_KEY_PATH`	`your-provate-key-path`
	`SAML_ACS_URL`	The externally visible assertion consumer service address, for example `https://scoutboard.example.com/api/v1/auth/login-saml/callback`. Set this only when UniconOS Management Board is published under a different address or port than it sees itself, for example behind a reverse proxy — by default, UniconOS Management Board builds this address from `HOST_NAME`, `USEHTTPS` and `PORT`.

With Microsoft Entra ID and OpenID Connect (`ENTRAID_OIDC`), add the optional claim `auth_time` to the ID token in the app registration of UniconOS Management Board. Without this claim, UniconOS Management Board can't check how recently an administrator signed in, and it refuses to grant Software rights with the message "Your identity provider does not report when you signed in, so Software rights cannot be granted." SAML sign-in needs no change, see Software rights.

Note:

From 15.2609, a SAML sign-in must be one UniconOS Management Board itself started: it rejects a response sent straight from the identity provider, for example by launching UniconOS Management Board from an Okta app tile or a Microsoft Entra **My Apps** tile. Each rejected response is logged as a **Login rejected** event. When UniconOS Management Board is reached over HTTPS, it reacts by discarding the response and immediately starting a fresh sign-in of its own, so the tile still reaches UniconOS Management Board, just through an extra round trip. To avoid that round trip, point the tile at `https://<Scout Board address>/api/v1/auth/login-saml` instead, or set `SAML_ALLOW_IDP_INITIATED=true` (default `false`) to accept the tile's response directly.

A SAML response is also single-use from 15.2609 on — UniconOS Management Board refuses a replay of the same response, whether or not `SAML_ALLOW_IDP_INITIATED` is set. `GET /login-saml` no longer accepts `SAMLRequest` as a query parameter.

UniconOS Management Board also applies three timing limits to a SAML sign-in:

- Up to 30 seconds of clock difference between the identity provider and UniconOS Management Board is tolerated.
- A response is accepted for at most 10 minutes after the identity provider issued it.
- The user has 45 minutes to finish signing in at the identity provider, once they start.

A refused sign-in shows **Sign-in could not be completed** with the reason, a reference code and a **Sign in again** link.

Configuring session limits for UniconOS Management Board

UniconOS Management Board ends a session after 30 minutes without activity, and 12 hours after the sign-in at the latest. The administrator then signs in again. A token of the REST API is valid for 4 hours. To change these limits, set the following values in the file `.env.custom` of UniconOS Management Board, for example `JWT_EXPIRATION=1h`, and then restart the **Scout Board** service:

SETTING	DEFAULT	WHAT IT LIMITS
`JWT_EXPIRATION`	30 minutes	The time without activity after which a session ends
`SESSION_MAX_LIFETIME`	12 hours	The time from the sign-in after which a session ends at the latest
`REST_TOKEN_LIFETIME`	4 hours	The time for which a token of the REST API is valid, see Token lifetime

Setup never changes `.env.custom`, so your values stay after an update.

Configuring AD domains for UniconOS Management Board

The AD domain for the logon process is being used from the provided data during the installation process.

In the `.env` file, the configuration is available. If the domain changes or additional domains need to be added, these can be added manually.

On the UniconOS Management Board logon screen, users select one of these domains in the **Domain** field.

Example:

DOMAIN OPTION	DOMAIN CONFIGURATION
One domain	`DOMAINS=["production.mydomain.com"]`
Multiple domains	`DOMAINS=["production.mydomain.com", "test.mydomain.com"]`

Note:

From 15.2611, an update keeps your list in `DOMAINS`, in its order, and a repair doesn't change it. If the list doesn't contain the domain of the UniconOS Management Board server, the update adds this domain at the end. If `LDAPS_SERVERS` already names a domain controller at that position, or if `DOMAINS` isn't a list, the update keeps `DOMAINS` unchanged and writes a warning to the setup log. Earlier versions replaced the list with the domain of the server at every update. If an earlier update reset your list, enter your domains again once. If the UniconOS Management Board server moved to another domain, the old domains stay at the start of the list, so remove the domains that you no longer need.

Note:

From 15.2609, UniconOS Management Board resolves the domain controller for a sign-in only from its own configuration — `DOMAINS` together with `LDAPS_SERVERS`, or `AUTH_DOMAIN_MAP` — instead of from a domain-controller address sent with the sign-in request or listed on the sign-in page. This applies to 15.2609 only; 15.2603 and 15.2607.x still use an address sent this way.

A sign-in for a domain that is neither in `DOMAINS` nor in `AUTH_DOMAIN_MAP`, and is not a subdomain of one of them, is refused with HTTP 412 and the message **Unable to resolve domain. Please contact your administrator.** The single-tenant sign-in page only ever offers a `DOMAINS` entry, so this mostly affects REST API callers and multi-tenant installations (`TENANT_MODE=multi-tenant`) that take the domain from the user name. A subdomain of a configured domain, for example `child.corp.example.com` under `corp.example.com`, signs in with no extra configuration.

To add a domain for REST API or multi-tenant sign-in without exposing it on the sign-in page, add it to `AUTH_DOMAIN_MAP` instead of to `DOMAINS`: `AUTH_DOMAIN_MAP={"CORP":{"domain":"corp.example.com","ldapServer":"ldaps://dc01.corp.example.com"}}`. The key (`CORP` above) is the short domain name used in `CORP\user`. `ldapServer` is optional; a plain string value names the domain only. An update keeps `AUTH_DOMAIN_MAP`.

If neither `LDAPS_SERVERS` nor a domain's `AUTH_DOMAIN_MAP` entry names a domain controller, UniconOS Management Board finds the domain controllers through DNS from 15.2611 on, see Domain controllers found through DNS.

If you are using LDAPS,in the `.env` file, you can configure them:
`LDAPS_SERVERS=["ldaps://my.ldaps.server", "ldaps://my.other.ldaps.server"]`

Note

UniconOS Management Board now uses `ldaps://` only and validates the DC certificate chain (chain + hostname/SAN). Previously, this validation was turned off. LDAPS silently failed and used plaintext `ldap://`.

Affected: On-Premises `AUTH_TYPE=AD` deployments with an untrusted, expired or SAN-mismatched DC certificate, or a plaintext `ldap://` entry in `LDAPS_SERVERS` / `AUTH_DOMAIN_MAP`.

Logins that worked in UniconOS Management version 15 2605 will fail. SSO modes and UniconOS Management Cloud deployments are not affected. The login screen will provide information on the cause, and the details will be in the `backend.log` log file.

Action required:

Untrusted certificate: From 15.2611, UniconOS Management Board checks the LDAPS certificate against its built-in root certificates and the certificates in the Windows certificate store of its server. On a domain-joined server, the certificate authority of your Active Directory is usually in this store already, so a domain controller certificate from your internal certificate authority works without further configuration. If the certificate authority is missing, import its root certificate with `certutil -addstore Root <root.cer>`. If the domain controller sends only its own certificate, also import the issuing certificate with `certutil -addstore CA <issuing.cer>`. Then restart UniconOS Management Board. Instead, you can set `LDAP_CA_PATH=<PEM CA bundle>` in the UniconOS Management Board `.env` file. `LDAP_CA_PATH` replaces the built-in root certificates and the Windows certificate store. It doesn't add to them.

> **Name mismatch:** In the UniconOS Management Board `.env` file, use the DC FQDN in `LDAPS_SERVERS` / `AUTH_DOMAIN_MAP`, or `LDAP_TLS_MODE=verify-ca`

> **Temporary workaround only:** In the UniconOS Management Board `.env` file, use `LDAP_TLS_MODE=insecure`.

> This is not supported in production environments, and it does not restore plaintext. (`ldap://` is refused in every mode)

> The default is `LDAP_TLS_MODE=verify-full`

> After you change the `.env` file, restart the Windows service **UniconOS Management Board**.

For how scripts sign in through the REST API and how long their token stays valid, see Using the REST API.

Domain controllers found through DNS

From 15.2611, if neither `LDAPS_SERVERS` nor the domain's `AUTH_DOMAIN_MAP` entry names a domain controller, Scout Board looks up the domain controllers of the domain in DNS, in the record `_ldap._tcp.dc._msdcs.<domain>`. It connects to each domain controller by its own name, so the certificate check of the default `LDAP_TLS_MODE=verify-full` passes. If a domain controller doesn't answer, UniconOS Management Board tries the next one. If none of them works, UniconOS Management Board connects to `ldaps://<domain>`, as earlier versions do. With `verify-full`, this connection fails if the domain controller's certificate names only its own host name.

A DNS answer isn't authenticated, so a forged answer could send the sign-in, including the password, to another computer of the domain that has a valid certificate. Therefore, UniconOS Management Board accepts a domain controller found through DNS only if its certificate shows that it is a domain controller of this domain. The certificate must have the KDC Authentication usage, as the Kerberos Authentication certificates of Active Directory Certificate Services have, or it must name the domain itself. UniconOS Management Board skips a domain controller whose certificate shows neither, for example one with a certificate from the legacy Domain Controller

template, and `backend.log` names the reason. This check is always on with `verify-full`. With `LDAP_TLS_MODE=verify-ca` or `LDAP_TLS_MODE=insecure`, it doesn't apply.

To skip the DNS lookup and this check, name the domain controller: set `ldapServer` in the domain's `AUTH_DOMAIN_MAP` entry to an `ldaps://` address with the domain controller's FQDN, for example `ldaps://dc01.corp.example.com`. For the server's own domain only, the first `LDAPS_SERVERS` entry also works.

Configuring the Database Layer key for UniconOS Management Board

UniconOS Management Board no longer fetches the Database Layer's public key at startup. Instead, it reads `ZERO_MQ_PUBLIC_KEY` from its `.env` file, which the installer writes.

- **UniconOS Management Board on the same server as the Database Layer:** no action needed. If `ZERO_MQ_PUBLIC_KEY` is missing or outdated, UniconOS Management Board takes the key from the local `dblayer.ini` instead.
- **UniconOS Management Board on a separate server from the Database Layer:**

  a. On the Database Layer server, copy the value of `[Certificates] DBLayerPublicKey` from `%PUBLIC%\Documents\Unicon\Scout\ScoutBoard\dblayer.ini`.

  b. On the UniconOS Management Board server, set `ZERO_MQ_PUBLIC_KEY=<key>` in the `.env` file and restart the **Scout Board** service. On a new installation, you can instead pass `UCPROP_SCOUTBOARD_REMOTE_DBLAYER_PUBLIC_KEY=<key>` to the installer; an upgrade keeps whichever key is already in `.env`.

  c. Repeat these steps whenever the Database Layer's key changes, for example after the Scout database is restored or UniconOS Management Board is re-pointed at a different Database Layer.

Without a usable key, UniconOS Management Board still starts, but cannot reach the Database Layer; its log names `ZERO_MQ_PUBLIC_KEY` as the setting to check.

UniconOS Management Board connects to the Database Layer only with encryption. It doesn't start if `ZERO_MQ_IS_CURVE_ACTIVATED` is set to `false`, and the error message names this setting.

If several Scout Servers share the Scout database, their Database Layers use the same key. Where `ZERO_MQ_PUBLIC_KEY` is set, it has the same value on every server.

From 15.2611, the Database Layer also needs to allow a UniconOS Management Board on a separate server, see Allowing a UniconOS Management Board on another computer.

Configuring the Database Layer client check

The Database Layer, the Windows service **Scout Board Database Layer**, connects UniconOS Management Board with the Scout database. UniconOS Management Board sends its requests to the Database Layer on port 22150 and receives events from it on port 22151. From 15.2611, the Database Layer checks the IPv4 address of every computer that connects to these ports. With the client check set to `Enforce`, the Database Layer accepts only the computers in its list of allowed hosts.

Setup turns the client check on as follows:

- **New installation:** setup sets the client check to `Enforce` and allows this computer (`self`). It allows other hosts only if you add them on the **Firewall** page or pass them in `DBLAYERALLOWEDCLIENTS`.
- **Update from 15.2609 or earlier:** setup sets the client check to `Enforce`. It allows this computer and the hosts that it finds when the update starts.

On an update, setup finds the computers that are connected to the Database Layer at that moment, and the addresses that the firewall rules of the Database Layer are limited to. The update page of the setup wizard shows the list, so that you can add computers that aren't running at that moment. A silent update uses the list as found. Afterwards, the Windows Application log contains a warning from setup that names the list.

If `dblayer.ini` already contains a client check setting, setup keeps it, also on an update.

Setup creates firewall rules for ports 22150 and 22151 only if you let it configure the firewall. With `Enforce`, setup limits these rules to the hosts in the list. With only `self` in the list, no other computer can reach the ports. An update never changes the firewall rules. If you use setup to switch the client check from `Enforce` to `Off` or `Audit`, setup opens these firewall rules to every address again. For the setup options, see Installing Scout Enterprise Management Suite and Unattended installation.

Client check settings

The client check settings are in the `[Network]` section of `dblayer.ini` on the Database Layer server, by default in `%PUBLIC%\Documents\Unicon\Scout\ScoutBoard\`. Edit the file in an editor that runs as administrator. The Database Layer reads the settings only when it starts, so restart the **Scout Board Database Layer** service after each change.

`DBLayerClientCheck` decides what the Database Layer does with a computer that isn't in the list:

VALUE	WHAT THE DATABASE LAYER DOES
`Enforce`	Refuses the computer before it can send a request.
`Audit`	Accepts the computer and logs it. Use this value to find out which computers connect before you switch to `Enforce`.
`Off`	Accepts every computer without a check. An empty value or a missing setting also means `Off`.

Any other value works like `Audit` and is logged as a warning at every start. Windows reads everything after the `=` as the value, so don't put a comment on the same line.

`DBLayerAllowedClients` holds the list of allowed hosts, separated by commas. The list can contain these entries:

- `self` stands for this computer: its loopback address and all its own IPv4 addresses. Keep `self` in the list. Without it, the Database Layer also refuses the UniconOS Management Board on this computer.
- IPv4 addresses, for example `10.0.0.20`.
- IPv4 address ranges, for example `10.0.0.0/24`.

Add every other computer that connects to the Database Layer, for example a computer that runs UniconOS Management Board or a Software Management backend for this Scout database. Host names and IPv6 addresses don't work: the Database Layer skips such an entry and logs it. Give the allowed computers static IPv4 addresses.

This example allows this computer and one UniconOS Management Board computer:

```
[Network]

DBLayerClientCheck=Enforce

DBLayerAllowedClients=self,10.0.0.20
```

If several Scout Servers use the same Scout database, the UniconOS Management Board of every server receives the events of all servers on port 22151. On each server, add the IPv4 addresses of

all other servers to `DBLayerAllowedClients`. A new installation allows only the server itself, and an upgrade adds only the servers that it finds connected. If a server refuses the UniconOS Management Board of another server, **Node health** in that UniconOS Management Board names the refusing server and `DBLayerAllowedClients`. For this setup, see Connections between the servers.

Allowing a UniconOS Management Board on another computer

A UniconOS Management Board on another computer can use the Database Layer only if the Database Layer server knows its key and, with the client check on, its IPv4 address. Setup registers the UniconOS Management Board that it installs on the Database Layer server itself. For a UniconOS Management Board on another computer, the last page of its setup shows the setting with its key, for example `/v"DBLAYERSCOUTBOARDKEYS=<key>"`.

To register the UniconOS Management Board:

1. On the Database Layer server, open `dblayer.ini` in an editor that runs as administrator.
2. In the `[Network]` section, add the IPv4 address of the UniconOS Management Board computer to `DBLayerAllowedClients`, for example `DBLayerAllowedClients=self,10.0.0.20`.
3. Add the key of the UniconOS Management Board to `DBLayerScoutBoardKeys`. If the setting holds keys already, separate the new key from them with a comma.
4. Save the file, and then restart the **Scout Board Database Layer** service.

UniconOS Management Board connects again by itself as soon as the Database Layer accepts it. You don't need to restart UniconOS Management Board.

Instead of editing `dblayer.ini`, you can run setup on the Database Layer server with the new values, in a command prompt that runs as administrator:

```
ScoutInstaller.exe /s /v"DBLAYERALLOWEDCLIENTS=self,10.0.0.20" /
v"DBLAYERSCOUTBOARDKEYS=<key>"
```

`DBLAYERSCOUTBOARDKEYS` adds the key to the keys that are registered already. `DBLAYERALLOWEDCLIENTS` replaces the whole list, so include `self` and every other allowed host. Setup adds the values, keeps the other settings and restarts the **Scout Board Database Layer** and **Scout Server** services, so plan a short interruption.

If you have limited the firewall rules **Scout DBLayer** and **Scout DBLayer Publish** to certain addresses, add the UniconOS Management Board computer to these rules as well.

Keep these points in mind:

- A UniconOS Management Board older than 15.2611 can't connect to the Database Layer of 15.2611. Upgrade UniconOS Management Board on the other computer first, and then register its key.
- Until its key is registered, the sign-in page of the UniconOS Management Board on the other computer shows "This UniconOS Management Board is not registered with the Scout server, so it cannot read any data. Ask your administrator to register its key; the UniconOS Management Board log names the key and the setting."
- If you don't have the key anymore, look in `backend.log`, the log file of UniconOS Management Board on the other computer. While the Database Layer refuses UniconOS Management Board, this log names the key to add.
- If you uninstall UniconOS Management Board on the other computer and install it again, it gets a new key. Register the new key.

Checking which computers connect

At every start, the Database Layer writes the mode and the list of allowed hosts to its log file `dblayer.log` and to the Windows Event Log, for example `DBLayer client check: Enforce, allowed clients: self, 10.0.0.20/32`. By default, `dblayer.log` is in the same folder as `dblayer.ini`.

- A refused computer appears in `dblayer.log` with the line `Refused a DBLayer client at <address>`. The Database Layer logs each address at most once an hour, and again after each of its restarts.
- With `Audit` or `Enforce`, the file `dblayer-clients.ini` next to `dblayer.ini` lists every computer that has connected. A computer that isn't in the list is marked as `unlisted`.

The client check doesn't hide the ports. The Database Layer still listens on all network interfaces, and a refused computer can still open a connection. Therefore, limit the firewall rules for ports 22150 and 22151 to the same hosts.

UniconOS Management Board HTTPS Certificate Setup

UniconOS Management Board HTTPS Certificate Setup

> **Note:**

> This certificate setup guide applies to **UniconOS Management Board version 15.2603 and later**

> This guide explains how to configure the HTTPS certificate that UniconOS Management Board presents to administrators in the web browser.

Three options at a glance

OPTION	WHEN TO USE	BROWSER WARNING?
Self-signed (default)	Lab, proof of concept, internal testing	Yes
CA-signed certificate from a public CA (DigiCert, GlobalSign, Let's Encrypt, ...)	Production, internet-facing	No
CA-signed certificate from your internal/corporate CA	Production, internal-only	No, on machines that trust your corporate root

Option A: Self-signed certificate (default)

If you do nothing, UniconOS Management Board generates a new self-signed certificate the first time the service starts. Browsers will warn that the connection is not trusted, but UniconOS Management Board itself works.

This is suitable for evaluation only. Use a CA-signed certificate for production.

Option B: CA-signed certificate, selected during installation

The UniconOS Management Board installer in 26.03 introduces a **Certificate** dropdown on the UniconOS Management Board settings page. The dropdown lists every certificate in the Windows **LocalMachine --> Personal** store that has an accessible private key. When you pick a certificate, the installer exports it together with any intermediate (and root) certificates it can find in the Windows certificate stores, and writes the result to `certificate.crt`.

STEPS

1. On the machine where you will install UniconOS Management Board, import your CA-signed certificate (with private key) into **LocalMachine --> Personal**. For PFX/PKCS#12 imports, use the Windows Certificate Manager (`certlm.msc`); the wizard places intermediate CA certificates into **Intermediate Certification Authorities** and any root into **Trusted Root**

Certification Authorities. Give your leaf cert a meaningful Friendly Name so you can recognise it in the dropdown.

2. Run the Scout bundle installer.
3. On the UniconOS Management Board settings page, open the **Certificate** dropdown and pick your imported certificate. If you imported it after the wizard was already open, click the refresh button next to the dropdown.
4. Continue the installation as usual.

> **Note:**
>
> If after install you see a browser certificate warning or you are redirected to a license-switch page when signing in, the chain in `certificate.crt` is most likely incomplete. The next section tells you how to verify and fix.

IF THE CERTIFICATE CHAIN IS INCOMPLETE AFTER INSTALL

The installer can only export intermediate certificates that exist in the Windows certificate store at the moment you run it. If you imported a leaf certificate without its accompanying intermediate(s) - a common situation when the source PFX did not bundle them - the installer will write only what it found, which may be just the leaf.

To diagnose:

1. Run the count check from Verifying your setup below. A count of 1 with a CA-signed certificate means the chain is incomplete.
2. Check `backend.log` for `certificate.crt contains only a leaf certificate` or `unable to verify the first certificate`.

To fix:

1. Stop the **Scout Board** Windows service.
2. Append the missing intermediate certificate(s) (and, for an internal/private CA, the root) to `<InstallDir>\scoutboard\certificate.crt`. The next section shows how to build the chain.
3. Start the **Scout Board** Windows service.

On the next install or upgrade, the installer will detect that you have built a complete chain and will leave your `certificate.crt` alone, so the manual fix survives.

Option C: CA-signed certificate, configured after installation

Use this when you cannot pick the certificate at install time, when the certificate is being renewed, or when you upgrade UniconOS Management Board from a release older than 26.03.

1. Stop the **Scout Board** Windows service.
2. Place your full chain file at `<InstallDir>\scoutboard\certificate.crt`. The default install directory is `C:\Program Files\Unicon\Scout\Scout Board\`.
3. Place your unencrypted `PKCS#8` private key at `<InstallDir>\scoutboard\private.key`.
4. Open `<InstallDir>\scoutboard\.env` in a text editor and set `CERTIFICATE=CUSTOM`. The legacy value `CERTIFICATE=SIGN` is still accepted as an alias for backward compatibility.
5. Start the **Scout Board** Windows service.

Building the certificate chain file

`certificate.crt` is a single text file in PEM format. It can hold one or many certificates back-to-back. The order matters: leaf first, then each intermediate moving up to the root.

A correctly built file looks like this:

```
-----BEGIN CERTIFICATE-----

... your leaf certificate (the one issued for your Scout Board hostname)

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

... intermediate CA certificate (the one that signed your leaf)

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

... next intermediate, if your CA uses a multi-level chain

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

... root CA certificate — only required if you use an internal/private CA
```

```
-----END CERTIFICATE-----
```

What to include depends on your CA:

1. **Public CA** (DigiCert, GlobalSign, Let's Encrypt, etc.): leaf + intermediate(s). The root is optional - UniconOS Management Board already trusts public roots out of the box.
2. **Internal/corporate CA**: leaf + intermediate(s) + your private root. Without the root, UniconOS Management Board cannot validate the chain because it does not consult the Windows trust store by default.

Most public CAs publish a downloadable bundle file (often called "intermediate bundle" or "CA bundle") that you can simply append to your leaf certificate using any text editor or:

```
# Build certificate.crt = leaf + intermediate bundle

Get-Content leaf.crt, intermediate-bundle.crt | Set-Content
certificate.crt
```

Verifying your setup

After placing the files and (re)starting the Scout Board service, run these checks.

HOW MANY CERTIFICATES DOES `CERTIFICATE.CRT` CONTAIN?

```
`Select-String -Path "C:\Program Files\Unicon\Scout\Scout
Board\certificate.crt" -Pattern "BEGIN CERTIFICATE" | Measure-Object`
```

If the count is `1` and you are using a CA-signed certificate, the chain is incomplete. Append your intermediates and restart the service.

DOES WINDOWS AGREE THE CHAIN IS VALID?

```
certutil -verify -urlfetch "C:\Program Files\Unicon\Scout\Scout
Board\certificate.crt"
```

DOES THE UNICONOS MANAGEMENT BOARD LOG SHOW ANY CHAIN WARNINGS?

Open `backend.log` in `C:\Users\Public\Documents\Unicon\Scout\ScoutBoard\` and look for either of these lines:

```
certificate.crt contains only a leaf certificate without the intermediate
CA chain

License validation error ... unable to verify the first certificate
```

Either of these means the chain is incomplete. The companion file `backend_error.log` in the same folder records error-level entries only - chain warnings appear in `backend.log`.

```
Select-String -Path "C:
\Users\Public\Documents\Unicon\Scout\ScoutBoard\backend.log" -Pattern
"leaf certificate|unable to verify"
```

OPEN UNICONOS MANAGEMENT BOARD IN A BROWSER

The browser should display the padlock without warnings, and you should be able to log in without being redirected to a license-switch page.

Troubleshooting

SYMPTOM	MOST LIKELY CAUSE	FIX
Browser warns "Not Secure" / "Certificate not trusted"	You are using the default self-signed certificate, or your `certificate.crt` is missing intermediates	Switch to a CA-signed certificate, or append the missing intermediates and restart the service
After login you are redirected to a license-switch page	`certificate.crt` does not contain the full chain (typically the installer could not find your intermediates in the Windows store)	Append intermediates to `certificate.crt` using Building the certificate chain file above and restart the service
Internal/corporate CA: redirect persists even with intermediates included	Your private root is not included in `certificate.crt`	Append your corporate root certificate to `certificate.crt` and restart the service
Browser warning resolves but UniconOS Management Board still does not load	The intermediate bundle includes the wrong intermediate, or its order is wrong	Verify with `certutil -verify -urlfetch`; the leaf must be first, intermediates above it in issuance order

Permissions, ownership and restart

1. Both `certificate.crt` and `private.key` must be readable by the account running the **Scout Board** Windows service. By default this account is `LocalSystem`, which has access.
2. Replacing either file requires a service restart - UniconOS Management Board reads them once at startup.
3. The installer cleans up both files on a complete uninstall, but preserves them across upgrades. Your custom certificate survives a UniconOS Management Board upgrade.
4. If you have manually edited `certificate.crt` to append intermediates, the installer detects this on upgrade and leaves the file alone - your manual fix survives.

Frequently asked questions

CAN I USE A WILDCARD CERTIFICATE?

Yes. Wildcard certificates work the same way — the leaf is the wildcard certificate, and the intermediate(s) come from the CA that issued it.

CAN I USE THE SAME CERTIFICATE AS ANOTHER SERVICE ON THIS MACHINE (IIS, CITRIX GATEWAY, ETC.)?

Yes, provided the certificate covers the hostname customers use to reach UniconOS Management Board. Import it into `LocalMachine\Personal` once and it will appear in the installer dropdown.

WHAT HAPPENS AT CERTIFICATE RENEWAL TIME?

Stop the Scout Board service, replace `certificate.crt` and private.key with the renewed file (full chain), then start the service. No installer rerun is needed.

WHAT HAPPENS DURING A UNICONOS MANAGEMENT BOARD UPGRADE?

Your existing `certificate.crt` and `private.key? are preserved across upgrades. You do not need to re-import or rebuild the chain file. If you previously appended intermediates manually, those are preserved too.

WHERE ARE THE UNICONOS MANAGEMENT BOARD LOG FILES?

`C:\Users\Public\Documents\Unicon\Scout\ScoutBoard\backend.log` (info-level entries and above, including chain warnings) and `backend_error.log` (error-level only) in the same folder. There is also a Windows service wrapper log at `%APPDATA%\ScoutBoard\`, but for certificate issues you want `backend.log`.

WHY DOES UNICONOS MANAGEMENT BOARD NEED THE INTERMEDIATES IN THE FILE RATHER THAN RELYING ON THE WINDOWS CERTIFICATE STORE?

The Scout Board service uses its own bundled trust list and does not read intermediate certificates from the Windows store at runtime. This keeps UniconOS Management Board behaviour consistent across servers regardless of what is installed in the local Windows trust store. The trade-off is that you must ensure `certificate.crt` is a complete chain.

Interface

Logging in

Open the UniconOS Management Board URL in your browser to reach the logon screen. The browser tab shows the Citrix® logo and the product name UniconOS Management. On the logon screen, the Citrix logo is on the left. The logon form on the right starts with **Welcome back**. **Help** at the top right opens the UniconOS Management Board documentation on docs.citrix.com in a new tab.

1. Enter your **Username** and **Password**.
2. If more than one domain is configured for this Scout Server, select the correct **Domain** from the list. If only one domain is configured, it is selected automatically.
3. Click **Log on**.

> **Note:**

> Logon via Active Directory is supported. Click the eye icon next to the password field to show or hide the characters you typed.

> If UniconOS Management Board is set up for sign-in through an identity provider, for example Microsoft Entra ID or Okta, the browser opens the sign-in page of the identity provider instead of this logon screen.

Session limits

UniconOS Management Board ends your session after 30 minutes without activity, and 12 hours after you signed in at the latest. It then shows **Your session has ended** with **Sign in again**. While

you upload a file to **Software**, the session stays active, but not longer than 12 hours. To change the limits, see Configuring session limits for UniconOS Management Board.

Main window

The UniconOS Management Board main window after logon shows the **Dashboard** with status information and an overview on different aspects of the managed devices.

From UniconOS Management 2611, UniconOS Management Board uses the Citrix colors, fonts and logos. At the top of the **Navigation Panel**, next to the Citrix logo, it shows the product name UniconOS Management. Most labels and messages that said eLux® or Scout before now say UniconOS or UniconOS Management, for example **UniconOS Command Scheduler**. The Scout components keep their names, for example Scout Server and Scout Board Database Layer in **System Status**. The pages of **Software** and some labels about it still say ELIAS, for example **ELIAS backend** in **System Status**.

UniconOS Management Board display languages: **English**, **German** and **Japanese**. UniconOS Management Board uses the browser's preferred language — the language at the top of the browser's language list — and reads it again each time UniconOS Management Board loads; it has no language menu of its own. If that preferred language is anything other than English, German or Japanese, UniconOS Management Board is shown in English, even if German or Japanese appears lower in the browser's list. With some browser languages, some common buttons, field labels and table controls appear in the browser's own language instead.

Some text stays in English in the Japanese interface, for example some server-side error messages and a few built-in list and table controls. UniconOS Management Board does not translate names and values that administrators enter, such as OU, device, application and command names. Device states, such as **Online**, are translated. The Scout installation program is not affected: it remains available in English and German only.

	INFORMATION
1	Navigation Panel
2	The section selected on the Navigation Panel is displayed.
3	Global search for data and navigation elements

	INFORMATION
4	Help, system status, notifications, and user profile. See Help, notifications, and profile below.
5	The **Views** panel of the Devices section. Its first three entries show your devices in different ways.
6	For the selected OU, the contained devices are displayed on the right.
7	For the selected device, the details are displayed on the right.
	Use the **Navigation Panel** on the left to choose the section you want to work with. Next to **Dashboard** and **Devices**, the **Navigation Panel** offers **Security & permissions**, **Preferences** and **Software**.

Note:

Close the **Navigation Panel** with to save space. Then only the section icons will be displayed.

To give a page more room, you can also hide the panel between the **Navigation Panel** and the page, for example the **Views** panel in **Devices**. This works in **Devices**, **Security & permissions**, **Preferences** and **Software**. Click (**Hide sidebar**) at the right of the panel title. While the panel is hidden, the same icon (**Show sidebar**) is at the left of the page title. Pointing to it opens the panel over the page, and the panel closes again when you move the pointer away. To show the panel next to the page again, click the icon. UniconOS Management Board remembers your choice for each section, also at your next logon.

Help, notifications, and profile

On the right of the title bar, four icons give you quick access to help, system status, notifications, and your own account:

- **Help** opens **About**, **Online help**, **License information**, and **Download log files**. **About** shows the product name **Citrix UniconOS Management** and the version, for example **Version 15 2611**. In UniconOS Management Cloud, the product name is **Citrix UniconOS Management Cloud**.

- The colored checkmark icon opens **System Status**, a live health check of the Scout Server, the Scout database, the Scout Board database layer, the Citrix License Server, device licensing, recovery settings and, if Software Management is installed, the Software Management backend (**ELIAS backend**). Each check shows a green checkmark when healthy, or a warning triangle with a short explanation when it needs attention. Click the refresh icon to run the checks again.

- The bell icon opens **Notifications**. If nothing needs your attention, it tells you that notifications are shown here as soon as they become available.

- The profile icon opens a menu with **Appearance** and with **Log off** followed by your name, which logs you off.

Devices

The Devices section offers three different views to show your devices:

- **All devices**: All devices are displayed in a list regardless of their organizational unit. The list can be sorted and filtered as desired.
- **OU structure**: The hierarchical OU structure is displayed as a tree view. All organizational units (in the following OUs) can be expanded and displayed.
- **Dynamic Device Groups**: Defined Dynamic Device Groups are displayed with the devices currently matching.

For a selected OU or Dynamic Device Group, on the right under Devices, the assigned devices and schedules are shown. To the right, the details of an OU, a Dynamic Device Group or a device may be displayed.

Software

If Software Management is installed and you hold the Software rights, the **Navigation Panel** also offers **Software**. There you keep the software packages of your devices in containers, and you build and sign the images that the devices install, see Software.

Dashboard

The Dashboard is the first page you see after logging on. It gives you a quick, at-a-glance overview of your device infrastructure, without having to open any device list.

The Dashboard shows five panels:

- **Device overview**: The total number of devices, broken down into **Online**, **Software update in progress**, and **Offline**.
- **Device status**: A more detailed breakdown, grouped into three areas:
  - Connectivity: how many devices are **online**, **UniconOS 7 compatible**, currently **updating**, or **offline**.
  - Activity: how many devices have **not been seen in a month** or **in a year**.
  - Compliance and connection type: how many devices have **compliant images**, are **connected internally** (through the corporate network), or **connected externally** (for example through VPN or Scout Cloud Gateway).
- **Device distribution**: The total number of devices, broken down into **Internal**, **External**, and **Reserved** (device profiles created in advance, before the device has ever connected).
- **License information**: The Citrix® License Server this Scout Server uses, its status (for example, reachable or too old), and your entitlement status with its activation and expiry dates. Click to change the license server name. For details, see Showing license information.
- **Image compliance**: The total number of devices, broken down into **Compliant**, **No image defined**, and **Non-compliant**, plus a chart showing how many devices are running each firmware image.

A firmware image is called compliant if it is an image that has been predefined under **Preferences** > **Predefined firmware configuration**. For further information, see Firmware preferences.

Devices / All devices

In this view, all devices of your infrastructure are displayed as a list, regardless of which OU they belong to.

By default, the list shows the **Name**, **Status**, **MAC address**, and **Image** columns. Click a column header to sort the list by that field; click it again to reverse the sort order. To add, remove, or reorder columns, see Changing List settings.

Use the filter function to narrow the list down to relevant devices, see Filtering devices.

If required, adjust the number of rows (devices) displayed per page.

Select a device to show its details on the right, or use the context menu at the end of its row for quick actions such as issuing a command or mirroring the device.

Devices / OU structure

The OU structure is organized hierarchically and may reflect the structure of your organization. OUs can contain devices at each level, but also other OUs. An OU can represent a site with its devices or group devices with the same details.

The advantage of having a hierarchical structure is that properties such as the device configuration and applications can be inherited from top to bottom. Actions can be applied to all devices in an OU, optionally including all devices in subordinate OUs, in one step. This applies, for example, to a command like **Restart**.

OU structure view

1	Click to expand an OU
	An expanded OU is indicated by
2	For each OU, the number of included devices, defined applications and schedules are displayed.
3	For a selected OU, the devices contained are displayed on the right in the [device list].

The device list on the right shows the devices located in the OU selected on the left. Devices located in subordinate OUs are not displayed.

The device list can be edited, sorted and filtered in the same way as in the **All devices** view, see Display devices. The **Actions** context-menu for the devices is also available, for example to issue commands or mirror a device.

OU details

Select an OU to show its details on the right:

- **Device count in this and subordinate OUs**, **Application count**, and **Schedule count**
- **OU ID**, **Name**, and **OU Path**
- Whether the OU currently inherits its configuration, applications, and update or keyboard/mouse settings from its parent OU (**Use parent configuration**, **Use parent apps**, and similar fields), or defines its own
- The **Info 1**, **Info 2**, **Info 3**, and **Note** fields you set for the OU

Additional functions of a device list within the OU structure

The following functions are not available in the **All Devices** view.

1	Show schedules
	For each OU, you may schedule commands which are displayed as schedules.
2	Manually add individual devices to an OU
3	Create a link for a selected device
	A device can be uniquely identified and shared via a device link.
	If no device is selected, the link is created for the OU.

Devices / Dynamic Device Groups

Dynamic Device Groups you already have in your database can be displayed and used in the UniconOS Management Board. For Dynamic Device Groups, the same Scout commands are provided as for OUs.

1	Show existing Dynamic Device Groups
2	Open **Actions** context-menu for a Dynamic Device Group
3	Select a Scout command

For the selected Dynamic Device Group, the currently matching devices are displayed on the right. On the far right, the Dynamic Device Group's details are displayed.

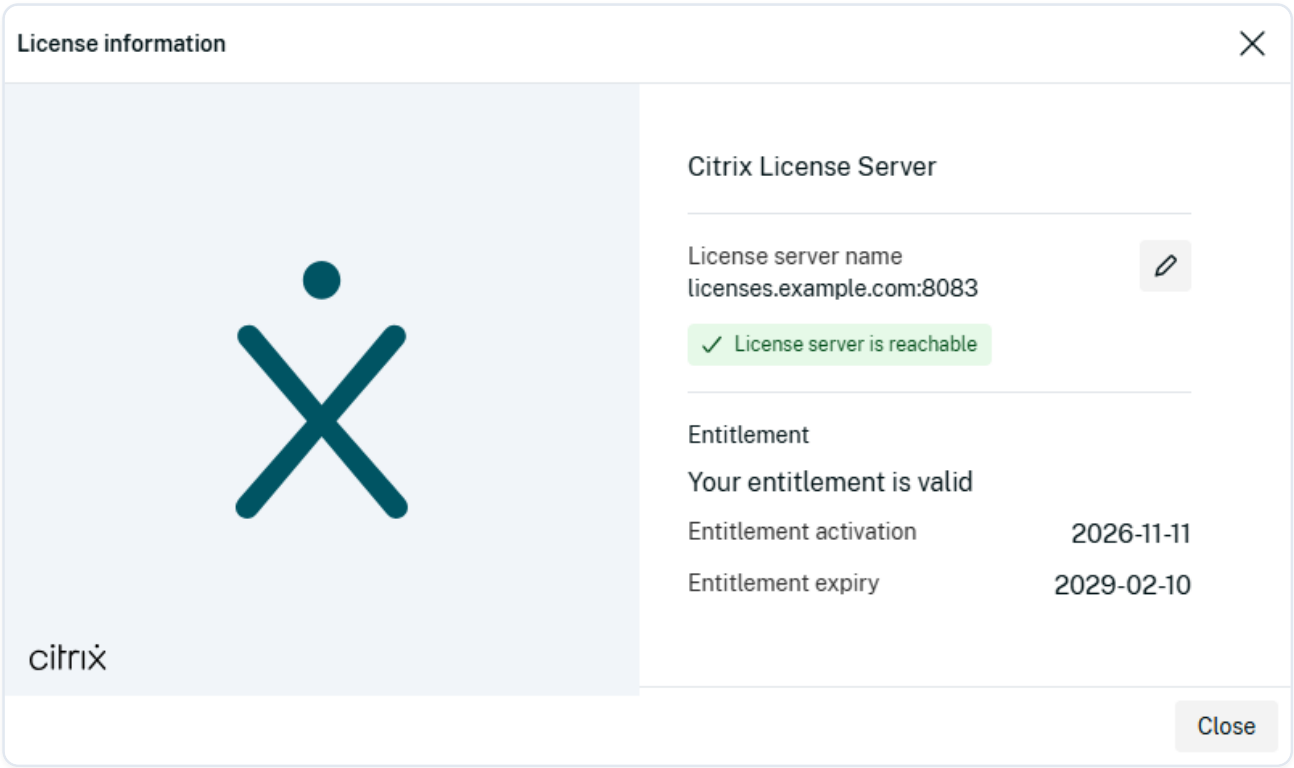
UniconOS Management Board shows a Dynamic Device Group only if you are allowed to see at least one of its devices. A command to a Dynamic Device Group needs your right for this command on every device in the group. If the group contains a device that you can't see, or on which you don't have this right, UniconOS Management Board refuses the command for the whole group. For details, see Rights on the devices and configurations that an action reaches.

Searching for devices and other objects

The UniconOS Management Board allows you to search for devices and OUs from any view. In addition, you can search for navigation elements like the **All devices** list and just jump to selected item.

Showing license information

The **License information** dialog shows which Citrix® License Server Scout uses, whether Scout Server reaches it, and whether your entitlement is valid. To open the dialog, click on the UniconOS Management Board title bar, and then click **License information**.



To change the License Server, click the pen icon next to its name. For what each status means and how the grace period works, see Licensing.

Devices and OUs

In the Devices section of the UniconOS Management Board, you can map the structures of your organization and manage the devices within OUs (organizational units). The OU structure makes it easier for you to both configure and manage your devices.

- The **device configuration** depends on the OU.

Different device configurations can be represented well by OUs. By default, the device configuration is inherited from the base - at the top of the OU tree - down through all existing OU levels to the devices. At each level, you have the option to break the inheritance. Even a single device can have an independent device configuration. For further information, see Device configuration.

- **Applications** are defined for OUs.

Application definitions are inherited from the top down, just like device configuration. At each level, you can decide whether you want to inherit applications from above -- or prefer to define applications dedicated to that level.

- Manage **schedules** and perform **commands**

Scout commands such as a device restart or a firmware update can be sent to all devices of an OU simultaneously. If commands are not to be executed immediately, they can be scheduled for a later time. The commands are then managed as schedules with an OU. You can also use schedules to define periodic execution of certain commands.

Alternatively, you can assign commands, schedules and other functions to selected devices in the **All devices** or **Dynamic Device Groups** views.

So, within the OU structure, inheritance applies by default: applications -- or more precisely application definitions -- are inherited by subordinate OUs. Device configurations are inherited down to the device level. When a new device is added to an OU, it automatically receives the application definitions and configuration of this OU.

For further information, see Device configuration.

Display devices

In UniconOS Management Board, the devices are displayed in the form of a list in all views. You can sort, filter and search a device list by any field. If possible, limit the scope of the displayed devices already in the first step by selecting the view:

All devices

- In the **Navigation Panel**, choose **Devices** > **All devices**.

All devices of your Scout infrastructure are displayed as a list.

DEVICES OF A SPECIFIC OU

1. In the **Navigation Panel**, choose **Devices** > **OU structure**.
2. Expand higher level OUs until you can see the relevant OU. To do this, click in front of an OU name.
3. Click the OU.

All devices of the selected OU are displayed. The device list of an OU has two tabs: One for the devices themselves and next to it all schedules of this OU.

DYNAMIC DEVICE GROUPS

1. In the **Navigation Panel**, choose **Devices** > **Dynamic Device Groups**.
2. Click one of the defined Dynamic Device Groups.

All devices matching the filter criteria of the Dynamic Device Group are displayed.

Device list functions

The same basic functions for a device list are available in all views.

In UniconOS Management Cloud, device lists also show certificate columns for enrolled devices, and the device menu offers to revoke certificates. See Enrollment settings for how devices are enrolled and how certificates work.

1. **Sort** by a field.
2. **Filter** by one or more fields.
3. : Define which of all available fields will be displayed as columns and choose the sequence.
4. Open the **Actions** context-menu of a device.
5. Show **details** of a device (or OU or Dynamic Device Group) on the right.
6. Selection column: Select multiple devices to apply an action on them. (from Scout 15 2302)

Especially when many devices are displayed, sorting and filter function are important tools.

List settings, filters and sorting are saved in the browser's local storage so that they are associated with a specific user and the browser used.

To re-use your settings beyond this and easily switch between settings , save them in individual views. For further information, see Individual device list views. (from Scout 15 2304)

Applied rules, Applied labels and Evaluated at

These three optional fields show which configuration rules and labels currently apply to each device. Add them with the list settings icon, the same way as any other field.

- **Applied rules** and **Applied labels** list every rule or label that currently applies to the device, in one cell. Click an entry to open that rule or label in a new tab. Hovering a rule shows the labels it assigns; hovering a label shows the rules that assign it.
- A struck-through entry is a deleted rule. A highlighted entry is a rule changed since the last evaluation. A grey entry is a deleted label. A dash means no rule or label applied.
- **Evaluated at** shows when Scout last evaluated the rules for the device. It is shown only while at least one rule applies to the device.

- Scout evaluates the rules, and updates all three fields, when the device requests its configuration from Scout.
- **Applied rules** and **Applied labels** can be filtered with **contains** or **is equal to**, and sorted; **Evaluated at** can be sorted but not filtered. A filter on these fields is refused when it would match rules or labels applying to more than 1,000 devices in total across your whole Scout infrastructure, not only the devices in the current list.
- These three fields are not included in the CSV export.

For the rule and label property list, see Dynamic device configuration.

Changing List settings

Specify which fields are displayed in the device list and in what order.

1. On the right, above the device list, click .

   All available fields are displayed in a list-field with scroll bar. The fields currently shown in the device list can be found at the top and are selected.
2. Use the scroll bar or search to view more fields in the list-field
3. To select specific fields for display, check the fields. To select all fields, click the icon in front of **Select all**.
4. To change the sequence, point to the left of a field name to . Drag the field to the desired position within the list-field.
5. Confirm with **Save**.

	[
1	Open list settings
2	Search for a field
3	Select a field to be displayed as a column
4	Drag a field to a new position (Sequence of columns)

Note:

If not all configured fields can be shown in the due to space constraints, a horizontal scroll bar will be displayed.

To use your beyond the current session, save them in a view. For further information, see Individual device list views. (from Scout 15 2304)

Sorting devices

-- applies for device lists in all views --

Note:

By default, a device list is sorted by the **Name** field.

1. In a , move the mouse pointer on a column header.

   A sorting arrow is displayed to the right of the field name.

2. Click the column header to sort the by this field in the direction specified. By default, sorting is in alphabetical descending order.
3. To sort in reverse order, click the column header again.

Once you have set sorting, the sorting arrow is permanently displayed next to the column header.

To re-use sorting settings and apply them to any device lists, save them in an individual view.

Note:

To undo a sort order, use the **Actions** menu of the column header.

Filtering devices

-- applies for s in all views --

Note:

> The filter function allows you to create multiple filter rules, which you can OR or AND.

1. Click the filter .
2. To define a filter rule (1), specify the relevant field (the column to filter by), an operator, and the value the field should have.
3. Click **Add rule** (2).

   The rule is added to the section below (4).
4. If you want to limit (AND) or extend (OR) the number of devices by a second filter rule, define another filter rule (1) and apply it via **Add rule** (2).
5. Make sure that the correct link type (OR/AND) is set (3).

   The link type applies to the linking of all filter rules and cannot be set differently for individual links.
6. Only if all the filter rules in the lower section (4) are correct, click **Apply filter**.

Only devices that match the filter rules are displayed. The **Filter** button is highlighted.

To re-use filter settings and apply them to any device lists, save them in an individual view.

> **Note:**
>
> To remove a filter, click next to the **Filter** button and select **Remove filter**.

FILTER OPERATORS

The following operators are available:

DATA FIELD TYPE	OPERATORS	EXAMPLE RULE
String (text)	`contains`, `is equal to`, `is not equal to`, `begins with`, `is empty`, `is not empty`	Name `contains` main
Numeric (numbers)	`=`, `>`, `>=`, `<`, `<=`	Flash size (GB) `<=` 4

DATA FIELD TYPE	OPERATORS	EXAMPLE RULE
Date	`on`, `is after`, `is before`	Last contact `is before` 05/02/2023
Logical values (Boolean)	`is`	Encrypted setup partition `is` No

Individual device list views

-- applies for s in all views -- (from Scout 15 2304)

List settings such as your selected fields for display as well as sorting and filtering criteria you have applied to a device list can be saved in views for re-use. You can save and reload as many views as you like. Use descriptive names for them. Views are saved to your user profile in the database.

Once saved, a view is available in all device lists, including the OU structure.

> **Note:**

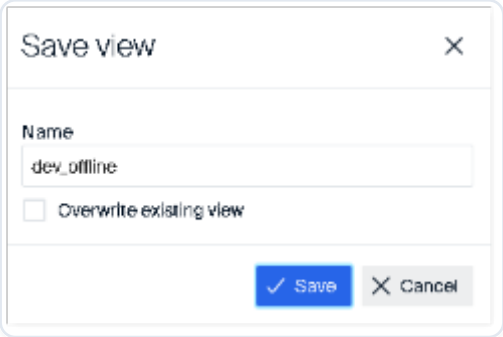
> Saving, renaming or deleting a view requires the **Create, edit and delete database reports** right. Administrators created in UniconOS Management Board without full access hold this right by default. Without it, saving a view shows **The view could not be saved. Please try again.**

1	Active view
2	Save, load, or edit views
3	Active filter rules
4	List settings

SAVING A VIEW

1. Apply the List settings, sorting or filter criteria as desired.
2. On the right above a device list, click ⋮

3. On the sub-menu, select **Save view**.



4. In the dialog, enter a name for the view and confirm with **Save***.

The new view's name is shown as the active view above the device list.

LOADING A VIEW

1. Click ⋮
2. On the sub-menu, select **Load view**.

   All views you have saved in this browser will be displayed.
3. Choose one of the displayed views.

The selected view is applied to the device list and will be shown as the active view.

EDITING A VIEW

-- Renaming or deleting saved views --

1. Click ⋮
2. On the sub-menu, select **Edit view**.

   All views you have saved in this browser will be displayed.
3. To rename a view, click ✎ to the right of it
4. To delete a view, click 🗑 to the right of it.

CHANGING A VIEW

1. Load the relevant view.
2. Change the List settings, sorting or filter criteria as required.
3. Save the view with the following options in the **Save View** dialog

   - Enter the name of the active view (identical spelling).
   - Select **Overwrite existing view**.

Adding devices

-- applies to device lists in the in OU structure --

Within an OU, you may add individual devices manually to the device list.

1. Display the OU structure, see Display devices.
2. Click to select an OU.
3. Above the device list, click **Add Device**.

   For the device name of the new device, the string `DEVICE-` and an appended incremented digit string are used by default.
4. If desired, replace the suggested device name.
5. Specify the MAC address of the new device.

> **Important**
>
> Use the following format: `00:11:22:33:44:55`

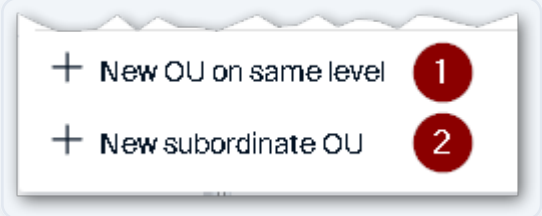
The new device is added to the current OU. It receives a unique Client Identifier.

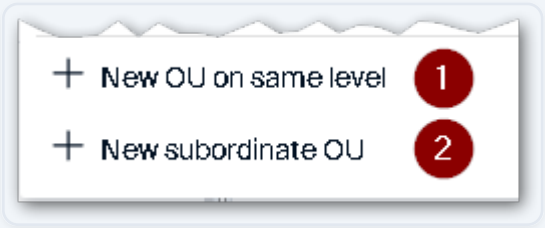
Adding OUs

-- applies to device lists in the OU structure

You can add new organizational units (OUs) in the OU structure at any level.

1. Display the OU structure, see Display devices.
2. To the right of an OU, click ⋮. From the context-menu, choose one of the following actions:

	
1	The new OU is created "next to" the selected OU.

	
2	The new OU is created below the selected OU.

3. Enter a name for the new OU and optionally additional information (**Info 1**, **Info 2**, **Info 3**) which will be shown in the device details (if configured).

Actions for devices

-- applies to device lists in all views --

Via a device list you have access to your devices and can, for example, send commands. All available actions can be found in the **Actions** context-menu of a device.

Performing an action for a device

1. To the right of a device, click ⋮.
2. From the context-menu, choose an action.

1	Execute or schedule a [Scout] command on the device
2	Rename, delete or move the device
3	Mirror the device
4	Retrieve diagnostic data

For enrolled devices in UniconOS Management Cloud, the menu also has an **Enrollment** entry with **Revoke certificate**. It appears only while the device's certificate is active.

> **Note:**
>
> You may also use drag-and-drop operations to move devices between OUs.

Performing actions on multiple devices

1. Use the selection column on the left to select multiple devices (1).

   Once you have selected multiple devices, a button (2) is displayed above the list.
2. Click (2), and then select a command or action from the context menu.

For enrolled devices in UniconOS Management Cloud, the **Selection** menu also offers **Revoke certificates**, which revokes the active certificates of every enrolled device in the selection and skips the rest. See Enrollment settings for how devices are enrolled and how certificates work.

> **Note:**
>
> To remove the selection of devices, click the minus icon above the selection column.

Actions for OUs

-- applies to the OU structure --

Performing an action for an OU

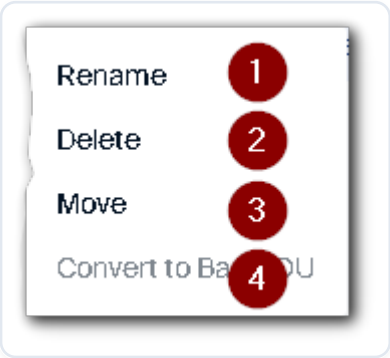
1. Display the OU structure, see Display devices.
2. To the right of an OU, click
3. From the context-menu, choose an action.

1	Execute or schedule a Scout command for all devices of the OU
2	Rename, delete or move the OU
3	Open the device configuration for this OU
4	Open the advanced device configuration for this OU
5	Add a new OU on the same level

6	Add a new OU below the current level

Renaming, deleting or moving an OU

1. To the right of an OU, click ⋮
2. From the **Actions** context-menu, choose **Edit**, and then one of the provided actions:

		
1	Overwrite the existing OU name	
2	Delete the OU	
3	In the dialog, browse the OU structure and select a new destination OU.	
4	Make this OU the base OU when appropriate rights provided	

Before any action is performed, you will be asked to confirm.

> **Note:**
>
> Renaming an OU to a name a sibling OU already has is refused, the same way creating or moving an OU already was.

Dynamic device configuration

(from Scout version 15 2601)

When an eLux endpoint connects to Scout, it will receive its configuration. By default, eLux endpoints are being managed within organization units.

Organization units are managed in a tree structure. While there is a base configuration for organization units and devices, with the tree structure, these device configurations are inherited down the tree structure.

Depending on the requirements, an administrator can break this inheritance and a different device configuration can be defined for an organization unit or for endpoints.

Changes made in an upper level in the tree structure will then not be inherited anymore.

Using a flag called "check independent configurations", the modified device configurations can be pushed down the tree structure, while manually choosing which configurations should be pushed.

When a device configuration has changed, there is a save option "Save and check independent configurations" for this feature to be applied.

For enabling a much more dynamic possibility to add specific configuration to devices, with the UniconOS Management Board version 15 2601, a rule and label based configuration approach has been added in addition to the default configuration mechanism.

Labels

A label is a small unit for defining specific configurations for a defined use-case. In a label, one or more sections from the device configuration can be added and the configuration can be defined.

This way, each label has a group of specific configurations, which can be applied to an endpoint.

One or more labels can be assigned to a defined rule.

The sequence of the labels also defines the priority or the configuration for an endpoint.

If the endpoint receives two labels, and both labels contain the same configuration section but with different configurations, the higher priority label will override the other during the merge of the configurations.

A label can also carry environment variables in its **Environment** section. Selecting **Encrypt value** for an entry stores the value encrypted and delivers it to the device encrypted, the same way as an

environment variable set directly on an OU or a device. A variable without **Encrypt value** reaches the device unchanged.

A label can also set the start scenario of devices, for example **Boot to VDI**, and turn on browser prompt suppression. For this, add settings of the **User authentication** section to the label: **Start scenario**, or **Suppress browser prompts** and **Trusted browser origins**. For details, see Start scenario and Labels and advanced file entries.

For Boot to VDI, the UniconOS desktop and kiosk mode, UniconOS Management Board also offers ready-made labels: the configuration templates. You copy a template, adapt it and link it to rules like a label, see Configuration templates.

LABEL SETTINGS ON THE DEVICES

From UniconOS Management 2611, devices get the label settings in this table as you set them. Earlier versions didn't send these settings to the devices, or they sent other values:

SECTION	LABEL SETTINGS
Network	WLAN, Mobile Internet and VPN profiles, the LAN connection with its passwords, and the notifications on failed 802.1X authentication
Drives	Network drives with their passwords and with **Use Active Directory authentication**
Printer	Printers with all their settings
Firmware	The **Provider** of a static proxy, and the **Password** for the server
Desktop	The settings under **Advanced** and **Shortcut keys**, the **Time format**, and background pictures set by URL
Power management	The **Working days**. A label without working days keeps the working days and hours of the device configuration.
Keyboard and Mouse	**Console switch**

SECTION	LABEL SETTINGS
User authentication	The SAML settings with the IdP certificate, the **OpenID client secret** and the **Secret** of Evidian
Mirroring	The **Password**
Security	The **User rights**
UniconOS Command Scheduler	Jobs with the preset time of the job dialog
Applications	The session options of RDP and VMware Horizon applications

A label changes only the settings that it contains. The network profiles, network drives and printers of a label are added to those of the device configuration. A label entry with the same name replaces the entry of the device configuration, and the LAN connection of a label replaces the LAN connection. So a label can't remove these entries, and it can't remove background pictures either. Only the host names and the jobs of the UniconOS Command Scheduler of a label still replace the whole list of the device configuration.

If you upgrade from an earlier version, check your labels. Devices now apply label settings that they ignored before.

Rules

A rule contains a logical expression, with different parameters and values. When a device connects to Scout, and this logical expression is valid, then this rule will be applied.

Scout checks the rules each time it builds the configuration of a device: when the device requests its configuration, for example when it restarts, and when Scout builds the configuration in advance. A rule gives the same result in both cases. Rules on the supplier, the serial number or the device type already apply to the first configuration of a new device.

Each rule will have one or more labels assigned to it.

The device will then get the configurations defined in the assigned labels.

If multiple rules apply for an endpoint, the higher priority rule and the labels assigned to the rule will override the other during the merge of the configurations.

Configuration rules offer more parameters than the OU filter rules: in addition to the device properties listed in User defined OU filters, configuration rules also offer **OU Path**, **OU Name**, **OU ID**, **Device Name (Server)**, **Info 1** to **Info 3**, **In Subnet (IP Check)** and **Custom Property**.

Priorities

Labels and rules have a priority. The higher priority wins and overrides the lower priority one.

When the configuration for a device is calculated, the different configurations will be merged by following a defined sequence of priority.

The priority is listed from low to high, while the higher will override the same configuration parameters.

- Base configuration --> lowest priority
- Inherited device configuration or Independent device configuration
- Advanced device configuration
- Label (Assigned directly to the OU)
- Label (Assigned directly to the Device)
- Independent configuration (Device)
- Rules --> highest priority

Configuration templates

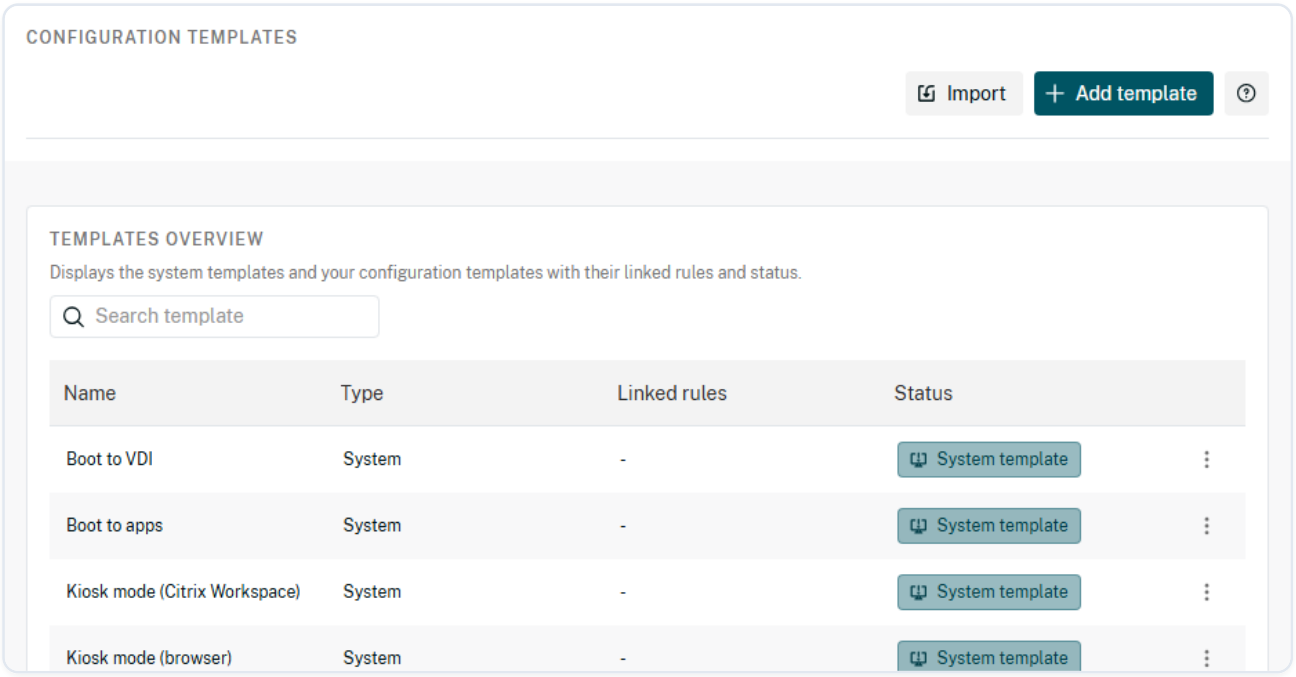
A configuration template is a ready-made configuration label for a common setup of devices, for example Boot to VDI or a kiosk. UniconOS Management Board comes with system templates, and you can add templates of your own. Like a label, a template has no effect on its own. You link it to a configuration rule, and the devices that meet the rule get all settings of the template. This way, you give a group of devices a complete setup without setting each value yourself.

For how labels and rules work, see Dynamic device configuration.

Opening the templates

In the **Navigation** menu, click **Devices**. In the **Views** panel, click **Device management**, and then **Configuration templates**. You see this entry only if you have the right **View labels**, see Rights.

Templates overview lists the system templates first, followed by the custom templates in alphabetical order. To find a template, type a part of its name or its type in **Search template**.



For each template, the list shows where it comes from, which rules use it and whether it is up to date:

COLUMN	SHOWS
Name	The name of the template.
Type	**System** for a template that comes with UniconOS Management Board. **Custom** for a template that an administrator added, copied or imported.
Linked rules	The configuration rules that link the template. A dash means that no rule links it. If the names don't fit into the column, point to them to see all of them.
Status	**System template** for a system template. **In use** if a rule links the template, and **Not in use** if no rule links it. **Outdated** if the system template that the template was copied from has changed since, see System templates.

The menu at the end of each row offers the actions for the template. Which items you see depends on the type of the template and on your rights:

MENU ITEM	DESCRIPTION
View	Shows the settings of a system template. To start a copy from there, click **Duplicate to edit**: it opens an unsaved copy in the template editor. Only for system templates.
Edit	Opens a custom template in the template editor.
Duplicate	Saves a copy of the template as a custom template with the same name. If a label or a custom template already has this name, UniconOS Management Board adds a number, for example **Boot to VDI (2)**.
Export	Downloads the template as a file, see Exporting and importing templates.
Delete	Deletes a custom template after you confirm. The item is unavailable for system templates and for templates that a rule links.

Above the list, **Import** creates a template from a file, and **Add template** opens an empty template in the template editor.

System templates

UniconOS Management Board comes with four system templates. Two of them set the start scenario of devices. The other two put devices in kiosk mode, with Citrix Workspace™ app or with the built-in browser:

TEMPLATE	WHAT IT SETS
Boot to VDI	The start scenario **Boot to VDI**: after the user signs in, devices open one Citrix® desktop or application in full screen.
Boot to apps	The start scenario **Desktop**, with an empty store and resource: devices start the

TEMPLATE	WHAT IT SETS
	UniconOS desktop, where users start their applications.
Kiosk mode (Citrix Workspace)	Kiosk mode with the Citrix Self-Service interface, for a device that several users share. You add the address of the store.
Kiosk mode (browser)	Kiosk mode with the built-in browser in full screen, without the navigation bar. You add the start page.

You can't change a system template. To use one, duplicate it and adapt the copy, see Using a template.

Each copy remembers the system template that it comes from. If an update of Scout brings a newer version of this system template, the copy shows the status **Outdated**. The template editor then says: "A newer version of the system template this template was copied from is available." The copy and the devices that use it don't change. To take over the new version, duplicate the system template again, adapt the new copy, and link it to your rules instead of the old copy.

BOOT TO VDI AND BOOT TO APPS

The **Boot to VDI** template sets the start scenario and nothing else. It has no store and no resource, so devices use the **Store** and the **Resource** of their device configuration, if it has any. To set a store and a resource for the devices of the rule, enter them in your copy. If you clear a field again, the value of the device configuration applies again.

While **Store** or **Resource** is empty, the template editor warns you, for example "Boot to VDI is selected but the store is empty. You can still save the template." For the sign-in types and the versions that Boot to VDI needs, see Start scenario.

The **Boot to apps** template sets the start scenario **Desktop** and empties **Store** and **Resource**. Use a copy to give the UniconOS desktop back to some devices, for example to a group of devices in an OU that uses Boot to VDI. If you remove the template from the rule, the devices get the start scenario of their device configuration again.

KIOSK TEMPLATES

The kiosk templates put devices in kiosk mode: after start-up, a device opens one application in full screen, and users can't start other applications. Each kiosk template has three configuration items:

- **Advanced file entries**: the entry `KioskMode` in the section `[Layout]` of `/setup/terminal.ini`. The value is `1` for **Kiosk mode (Citrix Workspace)** and `2` for **Kiosk mode (browser)**.
- **Files**: the file `/setup/kioskmode.ini` with the settings of the kiosk.
- **User authentication**: the start scenario **Desktop**, so that kiosk devices don't start Boot to VDI, even if their OU sets it.

The file of **Kiosk mode (Citrix Workspace)** sets up the Citrix Self-Service interface for a shared device:

```
[Parameters]

PreLogin=true

SharedUserMode=true

ShowLastUser=false

SelfSelection=false

LogoffMode=2

LogoffDelay=60

LogoffInfoTimeout=0

PassThroughMode=0



[Store1]

FriendlyName=

Url=

Default=true
```

With these settings, the kiosk works like this:

- Users sign in with their own account in a sign-in dialog before the store opens. As the file lists no domains, they type their domain in the dialog.
- The sign-in dialog doesn't show the last user.
- Users can't subscribe to more applications.
- The device signs users off 60 seconds after they close their last Citrix application or desktop, and it deletes their user data.

The devices need Citrix Workspace app and the other packages of the Citrix Self-Service kiosk. For these packages and all settings of the file, see Citrix Self-Service in kiosk mode.

The file of **Kiosk mode (browser)** opens the built-in browser in full screen with the start page. The browser shows no navigation bar, so users can't enter addresses:

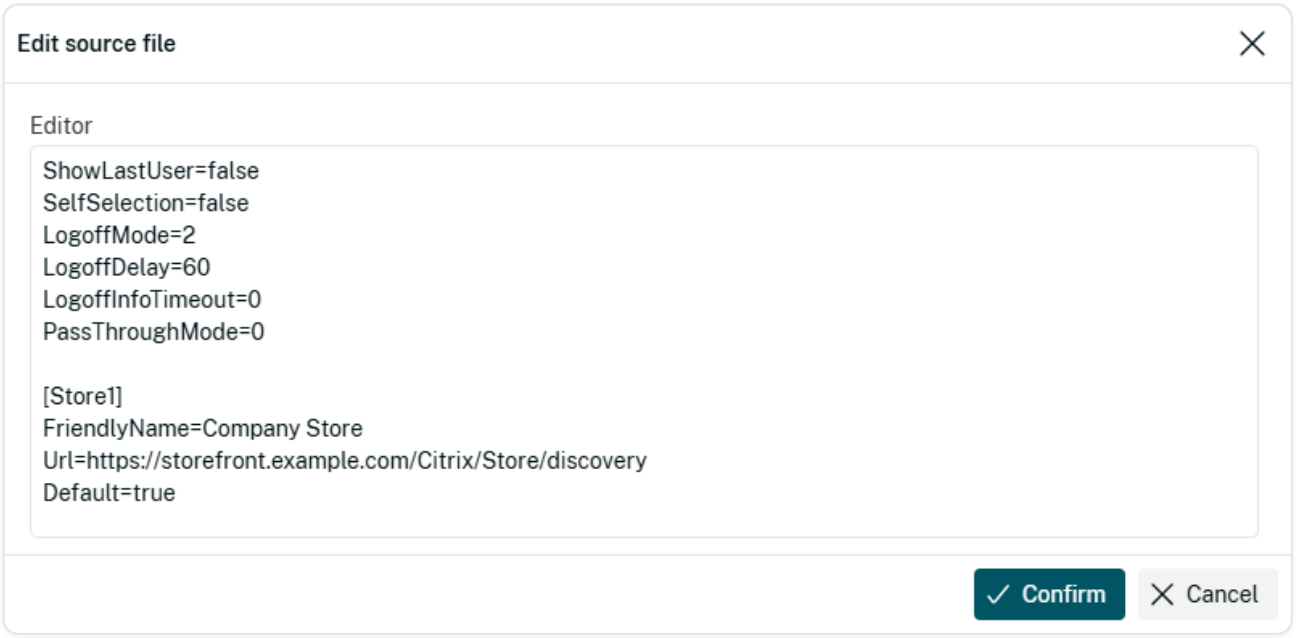
```
[Browser1]

Url=

Navbar=false
```

For all settings of the file, see Browser in kiosk mode.

Before you link a copy of a kiosk template to a rule, add the address that the kiosk opens. To copy a template, see Copying a system template. To add the address:

1. In **Templates overview**, open the menu at the end of the row of your copy, and click **Edit**.

   The template opens in the template editor.
2. Click **Edit**.
3. Under **Configuration items**, in the **Files** tile, open the menu at the end of the row of `/setup/kioskmode.ini`, and click **Edit source file**.
4. In **Editor**, enter the address after `Url=`: the URL of the Citrix store in `[Store1]`, or the start page in `[Browser1]`. To show a name for the store, enter it after `FriendlyName=`.
5. Click **Confirm**.
6. Click **Apply**.



Until the address is set, the template editor shows a warning, for example "Enter your store URL before you use this template". You can still save the template. Link it to a rule only after you have added the address.

Devices start the kiosk after the next sign-off or restart.

> **Important:**

> Don't add user names or passwords to `/setup/kioskmode.ini`, for example in a `[PassThroughCredentials]` section. Scout doesn't encrypt the content of files the way it encrypts passwords.

> To lock idle kiosk devices with a password, use the kiosk lock screen of the device configuration. A template can't set its password, see Kiosk lock screen.

Using a template

To give devices the settings of a system template, copy the template, adapt the copy and link it to a rule that matches the devices. You link a custom template that you added or imported in the same way.

COPYING A SYSTEM TEMPLATE

1. In **Templates overview**, open the menu at the end of the row of a system template, and click **Duplicate**.

UniconOS Management Board saves the copy as a custom template and shows "Template duplicated successfully".

2. Open the menu at the end of the row of the copy, and click **Edit**.

   The template opens in the template editor.

3. Click **Edit**.
4. Under **Template settings**, enter a name in **Label Name** that says what the template is for, for example the location of the devices. The name must differ from the names of all labels and templates.
5. Adapt the settings, for example the store and the resource of a Boot to VDI copy, or the address in the file of a kiosk copy, see Kiosk templates.
6. Click **Apply**.

LINKING A TEMPLATE TO A RULE

1. In the **Views** panel, click **Configuration rules**.
2. Open the menu at the end of the row of the rule that matches the devices, and click **Edit**. If no rule matches them yet, add one, see Rules.
3. Click **Edit**.
4. Under **Configuration Labels**, click **Add Label**.
5. In the **Add Label** dialog, select the check box of the template in **Select Labels**, and click **Confirm**. The list marks templates with **Template**.
6. Click **Apply**.

Scout evaluates the rules each time a device requests its configuration. The devices that meet the rule get the settings of the template with their next configuration update. To see which devices get the template, add the column **Applied labels** to a device list, see Display devices.

A template is a configuration label with a template mark. It also appears in **Configuration labels**, marked **Template**, and it works like any other label of the rule. If several rules or labels set the same value for a device, the one with the higher priority wins, see Priorities.

UniconOS Management Board doesn't refuse a template because of other templates, labels or settings. So make sure that each device gets a combination that works. For example, don't give a device Boot to VDI and a kiosk template at the same time, see Combinations to avoid.

Adding a template

To build a template of your own, start with an empty template:

1. Above the list, click **Add template**.

   The template editor opens with the tiles **Template settings**, **Common settings** and **Configuration items**.

2. Under **Template settings**, enter a name in **Label Name**. Unlike a label, a template has no color.
3. Under **Common settings**, set the values that the template gives the devices.
4. Under **Configuration items**, type a keyword in the search box, for example `Network`, and select a section or a single setting.

   The section gets a tile of its own. For a whole section, the tile shows the same settings as the device configuration. For a single setting, the tile shows only this setting.

5. Define the settings in the tile. Repeat the previous step and this step for each section that the template sets.
6. Click **Apply**.

A template needs at least one setting. Without one, UniconOS Management Board refuses to save it and shows "Label must have at least one configuration section or application assigned". To remove a section from a template, click the trash icon (**Remove section**) in its tile.

CONFIGURATION TEMPLATES / ADD TEMPLATE

Apply Discard

Boot to VDI is selected but the store is empty. You can still save the template.

Boot to VDI is selected but the resource is empty. You can still save the template.

TEMPLATE SETTINGS

Label Name

Boot to VDI - Sales

COMMON SETTINGS

Keyboard

Language

Not set

Language and region

Language

Not set

Time zone

Audio

Set volume

CONFIGURATION ITEMS

You can add sections or a single configuration entity by typing a keyword in the search box.

Search

USER AUTHENTICATION

Start scenario

Boot to VDI

Store

Enter the store URL

Enter a store. Without it, devices cannot start the virtual desktop.

Resource

Enter the desktop or application name

Enter a resource. Without it, devices cannot start the virtual desktop.

Devices with this configuration start only the selected resource. Other applications will not start on these devices. Requires Citrix Workspace app 2608 or later.

The common settings are values that many templates need. A setting that shows **Not set** isn't part of the template, so devices keep the value of their device configuration:

SETTING	TYPE	DESCRIPTION
Keyboard > **Language**	Dropdown	The keyboard layout of the devices.
Language and region > **Language**	Dropdown	The desktop language of the devices.
Language and region > **Time zone**	Dropdown	The time zone of the devices.
Audio > **Set volume**	Toggle	Adds the volume to the template. When you turn it on, **Volume** appears.
Audio > **Volume**	Slider	The volume of the analog, digital and USB output, from 0 to 100 percent. The initial value is 50 percent. Mute stays as the device configuration sets it.

Exporting and importing templates

You can save a template to a file and create a template from such a file, for example to use the template in another Scout installation.

To export a template, open the menu at the end of its row, and click **Export**. UniconOS Management Board downloads a JSON file named `template-<name>-<date>.json`. You can export system templates and custom templates. UniconOS Management Board leaves passwords and the values of encrypted environment variables out of the file.

To import a template:

1. Above the list, click **Import**.
2. In the **Import template** dialog, click **Select file**, and select a template file.

   The dialog shows the name of the template.

3. If the file contains a user-defined command for the **UniconOS Command Scheduler**, enter the **Device password**.
4. Click **Import**.

UniconOS Management Board saves the template as a custom template and shows the name that it saved the template under. If a label or a template already has the name, UniconOS Management Board adds a number. After the import, enter the passwords that the template needs.

UniconOS Management Board imports only template files that UniconOS Management Board exported. It refuses a file that a newer UniconOS Management Board version exported.

Rights

Configuration templates use the rights of configuration labels. You find them under **Dynamic Configuration** in **Base rights**:

RIGHT	ALLOWS
View labels	Seeing **Configuration templates** in the **Views** panel, opening templates and exporting them
Create labels	**Add template**, **Import**, **Duplicate** and **Duplicate to edit**
Edit labels	Changing custom templates
Delete labels	Deleting custom templates

To link a template to a rule, you also need the rights for rules, for example **Edit rules**. For all rights of dynamic device configuration, see Dynamic Configuration rights.

User rights

User rights is where you control which device configuration settings the person using the device is allowed to change themselves.

Opening User rights

1. Open the device configuration:
   - Base configuration: **Devices** > **Device management** > **Device configuration**.
   - OU or device: use its context-menu action **Open the device configuration**. See Actions for OUs or Actions for devices.

2. Go to **Security** > **User rights**.
3. Click **Edit**. For an OU or a device, turn off **Use parent device configuration** first.
4. Click an entry in the tree to switch it between allowed and locked.
5. Click **Apply** (or **Apply settings** for a single device) to save.

The tree mirrors the sections of the device configuration dialog, for example **Screen**. Each entry shows one of three states:

ICON	MEANING
Check mark	The device user is allowed to change this setting.
Cross	The setting is locked — the device user can see it but can't change it.
Minus sign	Some entries below this one in the tree are locked, others are allowed.

Search user right... filters the tree by name. The menu next to it offers **Allow all features** and **Lock all features** for the whole tree.

Modified user rights become active on the next device restart.

> **Note:**

> Administrators need the **Security** administrator permission to see this section, and **Edit user rights** to change it. These are administrator permissions, distinct from the device-user rights of the same names shown in the tree.

Screen rights

Under **Screen**, **Resolution** and **Rotation** sit next to **Manage monitors**, and **Manage monitors** expands into five entries of its own:

ENTRY	FOUND UNDER	WHAT THE DEVICE USER CAN NO LONGER CHANGE
Resolution	Screen	The resolution and refresh rate of each display. Locks the device's **Show all resolutions** option with it.
Rotation	Screen	The orientation of each display.
Manage monitors	Screen	Everything the five entries below it lock, plus Resolution and Rotation.
Set as primary	Manage monitors	Which display is the primary display.
Delete monitor	Manage monitors	Switching a display off.
Duplicate monitor	Manage monitors	Switching between **Extend displays** and **Mirror all displays**.
Scaling factor	Manage monitors	The display scaling factor.
Extend monitor	Manage monitors	How displays line up with each other in an extended desktop (**Vertical alignment** on the device).

These five individual entries, plus Resolution and Rotation, apply on devices running UniconOS 2609 or later. On earlier UniconOS versions, only **Manage monitors** has an effect, as a single on/off switch for all monitor settings, including arrangement.

Locking **Manage monitors** locks all five entries below it, and UniconOS Management Board then shows Resolution and Rotation as locked too. To lock only some monitor settings, leave **Manage monitors** allowed and lock individual entries instead. Locking all five entries below **Manage monitors** also locks **Manage monitors** itself, which locks Resolution and Rotation as well.

> **Note:**

> In a configuration label, locking **Manage monitors** does not lock **Resolution** and **Rotation** — unlike in the base configuration, an OU, or a device. Lock these two entries separately in a label if you need them locked too.

> Allowing any of these entries also allows **Manage monitors**, **Screen**, and the top-level **Device configuration** entry.

On UniconOS 2609 and later, none of these entries locks how displays are arranged relative to each other on the desktop.

On the device, a setting locked by these entries stays visible but greyed out.

Keyboard and Mouse rights

Under **Mouse/Keyboard**, four entries control the virtual (on-screen) keyboard:

ENTRY	FOUND UNDER	WHAT THE DEVICE USER CAN NO LONGER DO
Enable virtual keyboard	Mouse/Keyboard	Show the keyboard themselves. The **Accessibility** entry disappears from Quick System Access on the desktop, the logon screen and the lock screen, and the **Show on screen keyboard** switch disappears from the device configuration.
Virtual keyboard theme	Mouse/Keyboard	Change the keyboard's theme on the device.
Virtual keyboard layout	Mouse/Keyboard	Change the keyboard's layout on the device.
Prevent closing the virtual keyboard	Mouse/Keyboard	Change on the device whether the keyboard has a close button. The **Hide keyboard close button** switch disappears from the device configuration.

The first three entries apply on devices running UniconOS 2609 or later, and **Prevent closing the virtual keyboard** on devices running UniconOS 2611 or later. The four entries are independent of each other; locking **Mouse/Keyboard** locks all four. As everywhere in this tree, allowing one of these entries also allows **Mouse/Keyboard** and the top-level **Device configuration** entry.

A device user can still close an open keyboard with its own close button even while **Enable virtual keyboard** is locked; the device then has no control to reopen it until the device restarts. To remove the close button, turn on **Prevent closing the virtual keyboard** under **Keyboard and Mouse** > **Virtual keyboard** in the device configuration.

> **Caution:**
>
> On a device with no physical keyboard, if **Enable virtual keyboard** is locked and **Start virtual keyboard automatically** is off, device users have no way to bring up the keyboard, not even on the logon or lock screen. On such devices, leave **Enable virtual keyboard** unlocked. To keep the keyboard on their screen, also turn on **Start virtual keyboard automatically** and **Prevent closing the virtual keyboard**.

> See Keyboard and Mouse for the settings these rights control, and for the upgrade behavior when a configuration already locks **Mouse/Keyboard**.

Security rights

Under **Security**, the entry **Edit user rights** also covers **Trusted browser origins** of the **User authentication** section. **Suppress browser prompts** and **Trusted browser origins** have no entries of their own in the tree. If **Retain local device configuration (unlocked fields)** is turned on under **Preferences** > **Devices**, devices apply the list of trusted origins only while **Edit user rights** is locked. See Browser prompts.

Device configuration

Every OU and every device has a **device configuration**: the complete set of settings UniconOS Management sends to its UniconOS devices, from network and display settings to security and firmware options. By default, a device configuration is inherited from the base configuration down through the OU tree to the device. At each level you can break this inheritance and define independent settings.

To open the device configuration for an OU, use its context menu, see Actions for OUs. For a single device, use the device's context menu, see Actions for devices.

- **Device configuration** and **Advanced device configuration** are two separate tabs. This topic and its child topics cover **Device configuration**. For **Advanced device configuration**, see Dynamic device configuration.
- **Use parent device configuration** shows whether this OU or device currently inherits its settings. For an OU or a device, turn this off to define independent settings. The base configuration has no parent and no such switch.
- Click **Edit** to change settings, then **Apply** (base configuration or an OU) or **Apply settings** (a device) to save them. Changes reach devices with their next configuration update from Scout.
- **Search fields and values** filters the list of sections on the left to the ones containing a matching field name or value.
- The **Options** menu offers configuration-level actions such as exporting or importing a configuration.

The device configuration is organized into the following sections:

SECTION	COVERS
General	FollowMe Desktop
Persistent home	Keeping user settings and data across sessions and reboots
Network	LAN, WLAN, Mobile Internet and VPN connections
Display	Monitor layout, resolution and the screen saver
Security	User rights and application definitions
User authentication	The sign-in type, the start scenario (Desktop or Boot to VDI), the kiosk lock screen, logon screen options and browser prompts
Mirroring	Remote-viewing a device's session
Firmware	Firmware server settings, update reminders, and update notifications and battery limits
Multimedia	Audio output and input

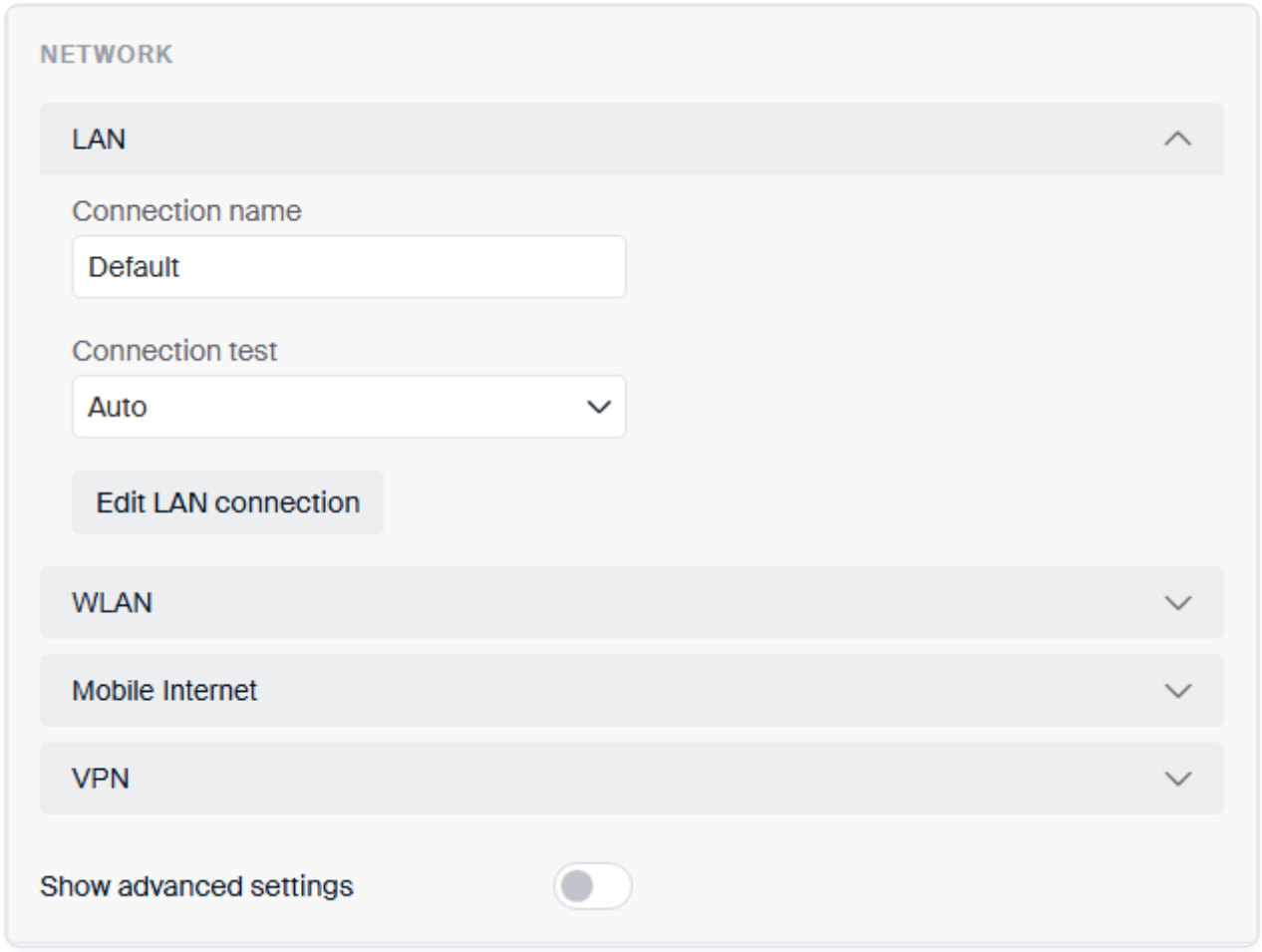
SECTION	COVERS
Desktop	Language, background, time, and the desktop information overlay
Printer	Defined printers and print services
Keyboard and Mouse	Keyboard and mouse settings, including the virtual keyboard
Hardware	USB settings and rules
Drives	Defined network and removable drives
Power management	Power profiles, low-battery notification and working hours
Diagnostics	Logging level and where diagnostic files are sent

Sections without a link are not yet covered by a dedicated topic.

Earlier versions had a **Kiosk Mode** section in the device configuration. It is removed, because devices never applied its settings. To set up kiosk mode, use a copy of a kiosk configuration template, see Configuration templates.

Device configuration / Network

The **Network** section defines how devices connect to networks. You find **Network** in the list of sections on the left of the device configuration. The section has four connection groups, **LAN**, **WLAN**, **Mobile Internet** and **VPN**. Below them, **Allow parallel connections** and **Preferred connection type** decide whether devices keep several connections up at the same time, see Parallel connections. The **Show advanced settings** switch adds settings that apply to all connections. For how these settings look on the device, see Network.



LAN

The **LAN** group holds a single connection. You name it, choose the connection test, and open its other settings with **Edit LAN connection**:

SETTING	TYPE	DESCRIPTION
Connection name	Text field	Name of the LAN connection. The base configuration names it `Default`.
Connection test	Dropdown	Each time the device connects, it checks whether it can reach the Internet. If it can't, it looks for a captive portal and opens it. **Auto**: the device tests WLAN connections, unless a central system proxy is defined. **On**: the device tests LAN and

SETTING	TYPE	DESCRIPTION
		WLAN connections. **Off**: the device runs no test.
Edit LAN connection	Button	Opens the **Edit LAN connection** dialog, see Connection settings.

WLAN

In the **WLAN** group, the **Connections** table lists the WLAN profiles with their **Name** and **SSID**. Click **Add connection** to create a profile. Above its groups, the **Add WLAN connection** dialog has these settings for the profile:

SETTING	TYPE	DESCRIPTION
Profile name	Text field	The name that the **Connections** table shows for the profile.
Connect automatically	Checkbox	Lets the device connect to this WLAN by itself. If you clear it, the user starts the connection from the live information on the system bar.
Connection test	Dropdown	Works as for the LAN connection, with **Auto**, **On** or **Off**, see LAN.

Below these settings, the dialog has the groups **Medium**, **IP**, **Advanced** and **IEEE 802.1X authentication**. They are described in Connection settings.

MOBILE INTERNET

For devices with a SIM card, a Mobile Internet profile defines how the device connects to a cellular data network. The **Connections** table lists the profiles with their **Name** and **APN**. Click **Add connection** to create a profile with these settings:

SETTING	TYPE	DESCRIPTION
Profile name	Text field	The name that the **Connections** table shows for the profile.
Connect automatically	Checkbox	The device connects to the cellular network by itself when the signal is strong enough.
APN	Text field	Access point name that the device uses to connect to the Internet over the cellular network, for example `internet.example.com`. The APN can contain only letters, digits, hyphens and dots, and it can have up to 100 characters. UniconOS Management Board refuses an APN with an empty part, such as `web..example.com`.
PIN	Text field	PIN of the SIM card, with 4 to 8 digits. If you leave it empty and the SIM card needs a PIN, the device asks for it at every connection.
Username	Text field	User name of the mobile account.
Password	Text field	Password of the mobile account.
MTU Size	Number field	Maximum transmission unit in bytes, from 1280 to 99999. The initial value is `1500`.
Roaming	Checkbox	Keeps the cellular data connection when the device

SETTING	TYPE	DESCRIPTION
		is outside the network of its mobile operator.

VPN

In the **VPN** group, you define profiles for virtual private network (VPN) connections. The **Connections** table lists the profiles with their **Name** and **Type**. Click **Add connection** to create a profile with these settings:

SETTING	TYPE	DESCRIPTION
Profile name	Text field	The name that the **Connections** table shows for the profile.
Connect automatically	Checkbox	The VPN client starts by itself and sets up the connection.
VPN client type	Dropdown	The VPN client that the device uses: **OpenVPN**, **Cisco AnyConnect VPN Client** or **Custom defined**. Use **Cisco AnyConnect VPN Client** for the Cisco Secure Client as well.
Configuration file	Text field	Appears for **OpenVPN**. Name of the OpenVPN configuration file without the file name extension. The file must already be on the device.
VPN client type number	Text field	Appears for **Custom defined**. ID number of your VPN client.
Advanced > **Type**	Dropdown	Proxy for the VPN connection: **No proxy**, **Manual** or **Auto**. The proxy

SETTING	TYPE	DESCRIPTION
		fields are the same as in Advanced.

The devices also need the VPN software packages and the matching certificates. For more information, see VPN.

PARALLEL CONNECTIONS

By default, a device uses either its LAN or its WLAN connection, never both at the same time. If you allow parallel connections, devices keep their LAN, WLAN and Mobile Internet connections up at the same time. A laptop on a docking station, for example, stays connected to the WLAN, so it can go on using it when you undock it.

You find the two settings below the **VPN** group:

SETTING	TYPE	DESCRIPTION
Allow parallel connections	Toggle	Keeps the connections of a device up at the same time. Off by default.
Preferred connection type	Dropdown	The connection that carries the network traffic while several connections are up: **Ethernet**, the default, or **Wi-Fi**. On the device, it is shown as **Active connection**.

If the active connection drops, for example when the LAN cable is unplugged, the device switches to a connection that is still up. When the preferred connection is back, the device switches back to it.

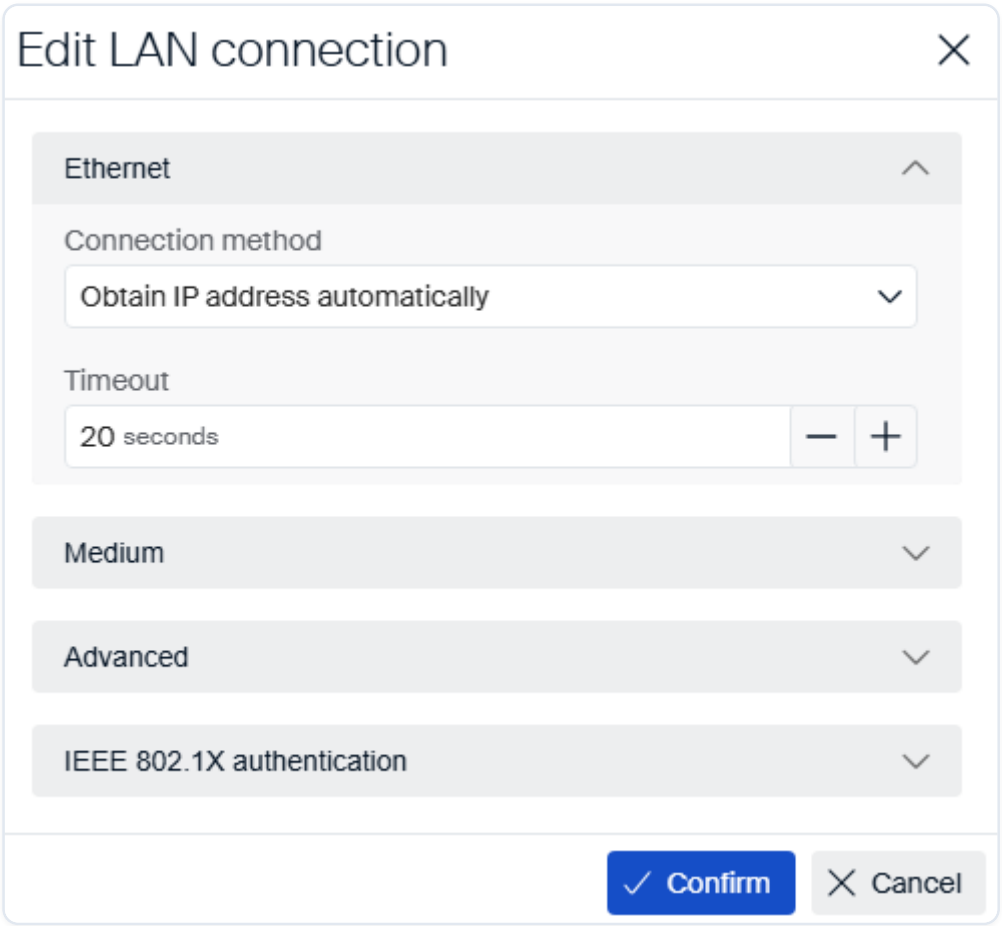
You can set both in the base configuration, for an OU, for a device and in configuration labels. Parallel connections need UniconOS 2611 or later on the devices. Devices apply a change after a restart. For what users see on the device, see Using Ethernet and Wi-Fi at the same time.

A device detects the sign-in page (captive portal) of a WLAN only once the WLAN is the active connection, for example after the LAN cable is unplugged.

In `/setup/terminal.ini`, Scout writes the settings as `AllowParallelConnections` and `PreferredMedia` in the `[Network]` section.

CONNECTION SETTINGS

The dialogs for LAN and WLAN connections have groups of settings. Open a group to see its settings. The LAN dialog has the groups **Ethernet**, **Medium**, **Advanced** and **IEEE 802.1X authentication**. The WLAN dialog has **Medium**, **IP**, **Advanced** and **IEEE 802.1X authentication**.



Ethernet and IP

These settings decide how the device gets its IP address and its other address data. In a LAN connection, they are in the **Ethernet** group. In a WLAN connection, they are in the **IP** group:

SETTING	TYPE	DESCRIPTION
Connection method	Dropdown	**Obtain IP address automatically**: the device gets its address from a DHCP server. **Use following IP**

SETTING	TYPE	DESCRIPTION
		address: you enter the address data.
Timeout	Number field	Appears for **Obtain IP address automatically**. Time in seconds that the device spends obtaining an IP address. The initial value is `20` for LAN and `60` for WLAN.
IP address, **Domain**, **Subnet mask**, **Gateway**, **DNS-Server 1**, **DNS-Server 2**	Text field	These fields appear for **Use following IP address**.

Medium

In a LAN connection, the **Medium** group sets the line speed:

SETTING	TYPE	DESCRIPTION
Speed	Dropdown	**Auto**, **100Mbit / Half duplex**, **100Mbit / Full duplex**, **10Mbit / Half duplex**, **10Mbit / Full duplex** or **1Gbit**.
MTU Size	Number field	Maximum transmission unit in bytes, from 1280 to 99999. The initial value is `1500`.

In a WLAN connection, the **Medium** group describes the wireless network:

SETTING	TYPE	DESCRIPTION
Network name / SSID	Text field	Name of the wireless network.
Timeout	Number field	Time in seconds that the device waits for the

SETTING	TYPE	DESCRIPTION
		connection. The initial value is `20`.
WLAN channel	Dropdown	**Auto** or a channel from 1 to 15.
MTU Size	Number field	Maximum transmission unit in bytes, from 1280 to 99999. The initial value is `1500`.
Encryption	Dropdown	Encryption of the wireless network: **None**, **WPA (PSK)** or **WPA2 (PSK)**.
PSK	Text field	Appears for **WPA (PSK)** and **WPA2 (PSK)**. Enter the pre-shared key, and choose **ASCII** or **Hexadecimal** for its format.
Hidden SSID	Checkbox	Select it if the network does not broadcast its name.
Access point MAC address / BSSID	Text field	Available when **Hidden SSID** is selected. If the network has several access points, separate their MAC addresses with semicolons. Devices need the BSSIDs to connect to a hidden network by themselves.

Advanced

The **Advanced** group decides whether the device uses the Scout options of the DHCP server, and which proxy server the connection uses:

SETTING	TYPE	DESCRIPTION
Ignore Scout server DHCP options (e.g. 222/223/224)	Checkbox	The device ignores the Scout options of the DHCP server.

SETTING	TYPE	DESCRIPTION
		Select it if you don't use DHCP options for Scout.
Type	Dropdown	Proxy server that this connection uses: **No proxy**, **Manual** or **Auto**. Browser applications that are set to use the system proxy use this proxy.
Proxy:Port	Text field	Appears for **Manual**. Address and port of the proxy server, for example `proxy.sampletec-01.com:3800`.
Proxy exception list	Text field	Appears for **Manual**. Addresses that the device reaches without the proxy. Separate several addresses with semicolons.
URL	Text field	Appears for **Auto**. Address of the proxy auto-config file, for example `http://proxy.sampletec-01.com/proxy.pac`.
Pass-through logon for proxy	Checkbox	The device authenticates at the proxy with the Active Directory logon data of the user.
Proxy username, **Proxy password**	Text field	User name and password for the proxy.

IEEE 802.1X authentication

In the **IEEE 802.1X authentication** group, you turn on 802.1X authentication for the connection and decide how often and how long the device tries:

SETTING	TYPE	DESCRIPTION
Activate authentication via IEEE 802.1X	Checkbox	Turns on 802.1X authentication for the connection.
Allow LAN connection without 802.1X if 802.1X fails	Checkbox	Lets the device connect after a timeout or an authentication error. If you clear it, the device connects only after a successful authentication. Available after you turn on 802.1X. In a WLAN connection, the setting reads **Allow WLAN connection without 802.1X if 802.1X fails**.
Number of auto-connect retries	Number field	How often the device retries the connection before it aborts. The initial value is `3`.
Number of authentication retries	Number field	How often the device retries the authentication before it aborts. The initial value is `4`.
Timeout authentication	Number field	Time in seconds before an authentication attempt is aborted. The initial value is `25`.

Note:

The WPA encryption uses the WPA supplicant and its configuration file `wpa.conf`. See Configuring WPA supplicant.

ADVANCED NETWORK SETTINGS

Turn on **Show advanced settings** to show settings that apply to all network connections, not to a single profile. The switch only shows or hides them, and its position is not saved. They control

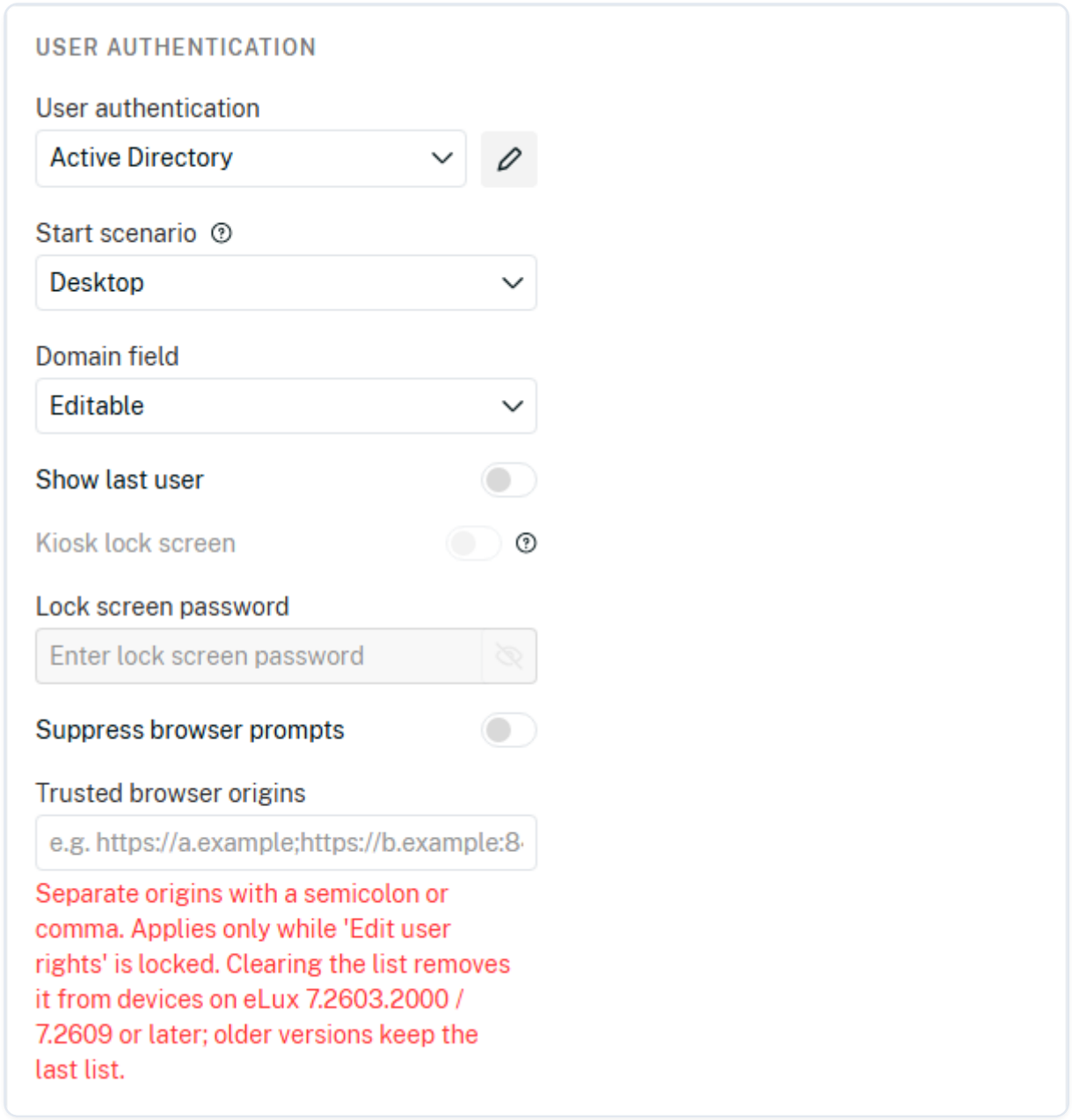
notifications, the DHCP host name, the connection to the Scout Server and host names that the device resolves itself:

SETTING	TYPE	DESCRIPTION
Show notification on Ethernet 802.1X fails	Checkbox	Shows a message on the device when the 802.1X authentication of a LAN connection fails.
Show notification on WLAN 802.1X fails	Checkbox	Shows a message on the device when the 802.1X authentication of a WLAN connection fails.
Ignore DHCP hostname (DHCP Option 12)	Checkbox	The device ignores the host name that the DHCP server sends in DHCP option 12.
When establishing a connection cancel after	Number field	Time in seconds after which the device aborts setting up its connection to the Scout Server. The initial value is `10`.
When connection is in idle state cancel after	Number field	Time in seconds after which the device ends an idle connection to the Scout Server. The initial value is `40`.
Send Keepalive packet every	Number field	Interval in seconds at which the device sends keepalive messages to the Scout Server. This needs the Scout Keep Alive service. The initial value is `600`.
Connections	Table	Host names that the device resolves itself, for networks without a DNS server. Click **Add connection** to open the **Add host** dialog. Enter the

SETTING	TYPE	DESCRIPTION
		address in **IP address** and one or more host names in **IP Name**. Separate several names with spaces. The table lists the names in the **Name** column and the address in the **IP** column.

Device configuration / User authentication

The **User authentication** section defines how users sign in on a device. You choose a sign-in type, enter its settings in a dialog, decide what the sign-in screen shows, and choose what devices start after sign-in. You can also stop browser prompts that interrupt users on the devices while they sign in to a store. You find **User authentication** in the list of sections on the left of the device configuration.



Note:

Administrators need the **Security** right to see this section and the **Edit user authentication** right to change its values, including **Start scenario**, **Store**, **Resource** and the browser prompt settings.

The settings of the section decide the sign-in type, what devices start after sign-in, what the sign-in screen shows, and whether a kiosk device locks. Two more settings stop browser prompts on the devices. Some settings are available only for certain sign-in types:

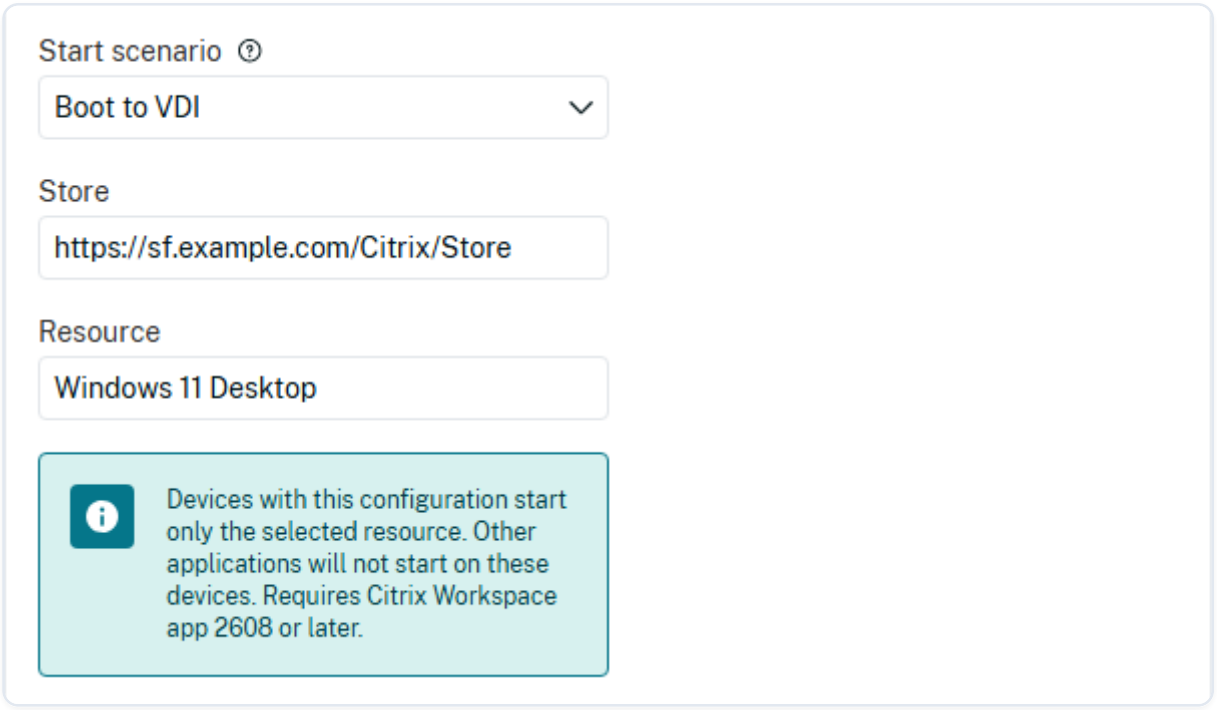
SETTING	TYPE	DESCRIPTION
User authentication	Dropdown	The sign-in type: **None**, **Active Directory**, **Active Directory + Smartcard**, **Evidian**, **Citrix®**, **OpenID Connect**, **Entra ID**, **Imprivata** or **SAML**. The pencil button next to the list opens the settings of the selected type. It is disabled for **None** and **Entra ID**.
Start scenario	Dropdown	What devices start. **Desktop**, the default, starts the UniconOS desktop. **Boot to VDI** starts one Citrix desktop or application in full screen after the user signs in. It isn't available for **None**, **Evidian** and **Citrix**. See Start scenario.
Store	Text field	Address (URL) of the Citrix store that offers the desktop or application. Appears for **Boot to VDI**. Enter up to 2,083 characters.
Resource	Text field	Name of the published desktop or application that devices start, as users see it in the store. Appears for **Boot to VDI**. Enter up to 255 characters.
Domain field	Dropdown	How the domain appears in the sign-in dialog. **Editable**: users can change the domain. **Not editable**: users see the domain but can't change it. **Hidden**: the dialog does not show the domain.

SETTING	TYPE	DESCRIPTION
		Available for **Active Directory**, **Active Directory + Smartcard** and **OpenID Connect**.
Show last user	Toggle	The sign-in dialog shows the name of the last signed-in user. Not available for **None**.
Kiosk lock screen	Toggle	Locks a shared kiosk device behind one password. Available for **None** only. See Kiosk lock screen.
Lock screen password	Text field	The password that unlocks the device. Available for **None** only.
Suppress browser prompts	Toggle	Stops browser prompts that interrupt the Citrix sign-in on the devices. Off by default. See Browser prompts.
Trusted browser origins	Text field	Websites, as https origins, on which browsers open Citrix links without asking, in addition to the stores of the devices. Separate them with a semicolon or a comma. Enter up to 2,048 characters. See Trusted browser origins.
Show information on the logon screen	Toggle	Shows a tile with device information on the sign-in screen. For the settings of the tile, see Logon screen information.

The sign-in types **None**, **Evidian** and **Citrix** don't draw the device information tile. If one of them is selected, UniconOS Management Board warns you about it. The tile settings are saved, but they have no effect on the device.

START SCENARIO

Start scenario decides whether devices start the UniconOS desktop or a Citrix virtual desktop. With **Boot to VDI**, a device opens one desktop or published application from a Citrix store in full screen after the user signs in. Users can't reach the UniconOS desktop, and other applications that you assign to the device don't start. When the session ends, the device returns to the sign-in screen. You find **Start scenario** directly below **User authentication**.



After the upgrade to UniconOS Management 2611, every existing configuration has the start scenario **Desktop**, so devices keep starting the UniconOS desktop. You can set the start scenario in the base configuration, for an OU or for a device, and in configuration labels.

Boot to VDI requires UniconOS 2609 or later and Citrix Workspace™ app 2608 or later on the devices. For what users see on the device, see Boot-to-VDI.

Sign-in types for Boot to VDI

Boot to VDI starts the virtual desktop only after the user signs in on the UniconOS sign-in screen, so it needs a sign-in type. It works with **Active Directory**, **Active Directory + Smartcard**, **OpenID Connect**, **Entra ID**, **Imprivata** and **SAML**. It doesn't work with the other three sign-in types:

- **None**: Users don't sign in on the device.
- **Evidian**: UniconOS doesn't support Boot to VDI with Evidian.
- **Citrix**: This sign-in type sets up a Citrix desktop lock of its own, see Citrix.

In the device configuration, UniconOS Management Board doesn't let you combine Boot to VDI with these three sign-in types:

- While **User authentication** is **None**, **Evidian** or **Citrix**, you can't select **Boot to VDI** in the **Start scenario** list. Point to the question mark next to **Start scenario** to see why: "Boot to VDI needs an authentication type other than None, Evidian or Citrix."
- While **Start scenario** is **Boot to VDI**, you can't select **None**, **Evidian** or **Citrix** in the **User authentication** list. The text below the list says: "None, Evidian and Citrix are not available while the start scenario is Boot to VDI."

Setting up Boot to VDI

Try Boot to VDI on a test device before you set it for many devices. If the store or the resource is wrong, the device can't start the virtual desktop, and users can't work on it. To set up Boot to VDI:

1. Set **User authentication** to a sign-in type that works with Boot to VDI, and enter its settings.
2. Set **Start scenario** to **Boot to VDI**.

   Store and **Resource** appear below the list.
3. In **Store**, enter the URL of the Citrix store.
4. In **Resource**, enter the name of the published desktop or application.

   With Imprivata XenDesktop® automation on UniconOS 2611 or later, leave **Resource** empty. The Imprivata policy then decides which desktop starts, and users with several desktops pick one in a desktop chooser, see Desktop Chooser.

Devices that use this configuration switch to Boot to VDI after they get the new configuration and restart. To switch them back to the UniconOS desktop, set **Start scenario** to **Desktop**, and restart them.

In `/setup/terminal.ini`, Scout writes the start scenario in the `[Security]` section: `ScenarioType` (`0` for **Desktop**, `1` for **Boot to VDI**), `ScenarioStore` and `ScenarioResource`. For Boot to VDI, Scout also sets `BootToVDIEnabled=True` in the `[WFClient]` section of `/setup/ica/wfclient.ini`.

To give Boot to VDI to a group of devices, add **Start scenario** to a configuration label, and assign the label to a rule that matches the devices. In the label, **Start scenario**, **Store** and **Resource** work as in the device configuration. A label without **Start scenario** leaves the start scenario of the devices as it is. You can also start from a copy of the **Boot to VDI** configuration template, see Configuration templates. For labels and rules, see Dynamic device configuration.

Combinations to avoid

UniconOS Management Board doesn't refuse a configuration, a label, a rule or an advanced file entry because of the start scenario. Scout sends the start scenario to the devices as configured. So make sure that each device gets a combination that works:

- **Sign-in type**: In a label, UniconOS Management Board disables **Boot to VDI** only if the label sets the sign-in type as well. Devices with **None**, **Evidian** or **Citrix** don't use Boot to VDI, even if a label or an advanced file entry sets it.
- **Store and resource**: If **Store** or **Resource** is empty, the field shows a warning, for example "Enter a store. Without it, devices cannot start the virtual desktop." You can still save the configuration.
- **Kiosk mode**: Don't set up kiosk mode on devices with Boot to VDI. A device isn't designed to run both, and Scout Board doesn't warn you. The kiosk configuration templates set the start scenario **Desktop**, so that kiosk devices don't start Boot to VDI that their OU sets.
- **Advanced file entries**: Entries for `ScenarioType`, `ScenarioStore`, `ScenarioResource` or `BootToVDIEnabled` in **Advanced device configuration** > **Advanced file entries** win over the start scenario settings, because Scout sends them afterwards. This also applies to the advanced file entries of labels. If you set up Boot to VDI with such entries before UniconOS Management 2611, they keep overriding the start scenario until you delete them.

KIOSK LOCK SCREEN

The kiosk lock screen locks a shared kiosk device with one password that an administrator sets, instead of leaving an idle session open. To turn it on:

1. Set **User authentication** to **None**.
2. Turn on **Kiosk lock screen**.
3. Enter a password in **Lock screen password**.

If you turn on **Kiosk lock screen** without a password, UniconOS Management Board warns you: "The lock screen is switched on but no password is set. Until you set one, affected devices will lock with no way to unlock them." To remove the lock, turn **Kiosk lock screen** off.

Before you use the kiosk lock screen, keep these points in mind:

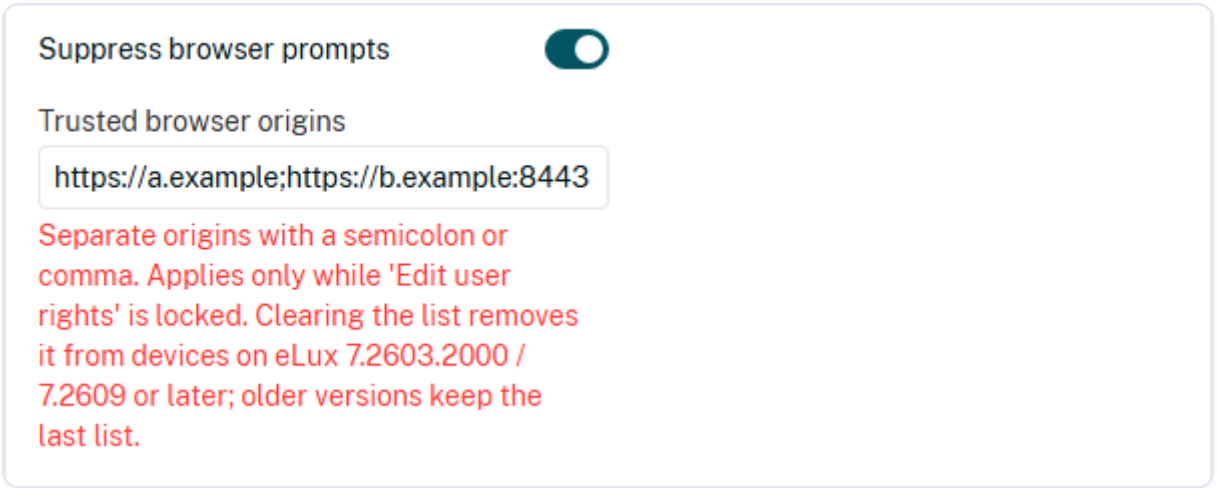
- All devices that use this device configuration share the password. It has no expiry and no complexity rules, so choose a strong password.
- The password can have at most 217 bytes in UTF-8 encoding. UniconOS Management Board refuses a longer password.

- A configuration label can't set the kiosk lock password. The kiosk lock password of the device configuration applies.
- The device applies the setting after a restart.
- The kiosk lock screen needs UniconOS 2609 or later on the device. Devices with an earlier version get neither the setting nor the password.
- The lock screen works only if the device already runs in kiosk mode, as **UC Self Service** or **UC Browser**. To set up kiosk mode, use a copy of a kiosk configuration template, see Configuration templates.
- If the device has an Imprivata bootstrap URL, the Imprivata lock screen replaces this one.

For the behavior on the device, see Kiosk lock screen.

BROWSER PROMPTS

Browsers can interrupt the Citrix sign-in on a device with prompts. Google Chrome and Microsoft Edge offer to save the password that the user has just entered. Chrome, Edge and Firefox also ask for permission before a website opens Citrix Workspace app, for example when the store detects Citrix Workspace app or when the user starts a published desktop or application. On kiosk and shared devices, users answer the same prompts again and again. **Suppress browser prompts** stops these prompts. You find it, together with **Trusted browser origins**, above **Show information on the logon screen**, for every sign-in type.



When you turn on **Suppress browser prompts**, the browsers on the devices change in two ways:

- Chrome and Edge don't offer to save passwords on any website, except Edge in a profile that is signed in with a Microsoft account. Both browsers also stop saving and filling in addresses and credit card data. Firefox on UniconOS never offers to save passwords, whatever the setting.

- On trusted origins, Chrome, Edge and Firefox open Citrix Workspace app without asking. Trusted origins are the stores of the device's StoreFront™ applications and the websites in **Trusted browser origins**. On other websites, the browsers still ask for permission.

Prompt suppression needs UniconOS 2609 or later on the devices. For what users see in each browser, see the **Browser Prompts** section of the UniconOS page Applications.

To turn on prompt suppression:

1. Turn on **Suppress browser prompts**.
2. In **Trusted browser origins**, enter the websites that open Citrix Workspace app but aren't stores of the StoreFront applications of the devices, for example your identity provider. For the format, see Trusted browser origins.
3. If **Retain local device configuration (unlocked fields)** is turned on under **Preferences** > **Devices**, check that the user right **Edit user rights** is locked. You find it under **Device configuration** > **Security** in **Security** > **User rights**, see User rights.

   The browser prompt settings have no user rights of their own. While devices keep the values of unlocked fields, they apply **Trusted browser origins** only if **Edit user rights** is locked.

Devices get the settings with their next configuration update from Scout. Each browser applies them the next time it starts. A browser that is already running keeps its previous behavior until the user closes all its windows and starts it again, or until the device restarts.

In `/setup/terminal.ini`, Scout writes **Suppress browser prompts** as `SuppressPrompts` (`true` or `false`) in the `[Browser]` section. Scout always sends this key, also when the setting is off. **Trusted browser origins** becomes `BrowserTrustedUris` in the `[Security]` section.

Trusted browser origins

The devices trust the stores of their StoreFront applications by themselves. In **Trusted browser origins**, add the other websites that open Citrix Workspace app, especially these:

- The identity provider, if users sign in to the store through it, for example with Microsoft Entra ID. With such a federated sign-in, the page of the identity provider opens Citrix Workspace app, not the store.
- Stores that users open with a browser application instead of a StoreFront application.

Enter each website as an https origin, which is its address without a path, for example `https://idp.example.com`. Separate several origins with a semicolon or a comma, for example

`https://a.example;https://b.example:8443`. The field takes up to 2,048 characters. UniconOS Management Board accepts an origin only if it follows these rules:

- It starts with `https://` in lowercase letters.
- Its host name has at least two parts, such as `example.com`, and contains only letters, digits, hyphens and dots. Enter international domain names in their ASCII form (punycode).
- It can end with a port number, for example `:8443`.
- It has no path, no wildcard such as `*` and no space inside. Spaces before or after an origin don't matter.

If an origin breaks a rule, the field turns red. The text below the field then starts with "Not accepted:" and names the refused origins. UniconOS Management Board doesn't save the section until you correct or delete them.

In Chrome and Edge, an origin also covers its subdomains, and an origin without a port covers every port. To trust one host only, put a dot before the host name, for example `https://.store.example.com`. Firefox trusts an origin only for exactly its host and port, so add the port if the website doesn't use port 443.

> **Important:**

> Devices also use **Trusted browser origins** for Kerberos authentication. Each origin in the list also allows automatic Kerberos sign-in to that host. Add only origins that you trust.

> If you clear **Trusted browser origins**, Scout stops sending the list. Devices with UniconOS 2609 or later, or with UniconOS 2603 LTSR CU2 or a later cumulative update, then delete the list that they got from Scout before. Devices with earlier versions keep the last list.

Labels and advanced file entries

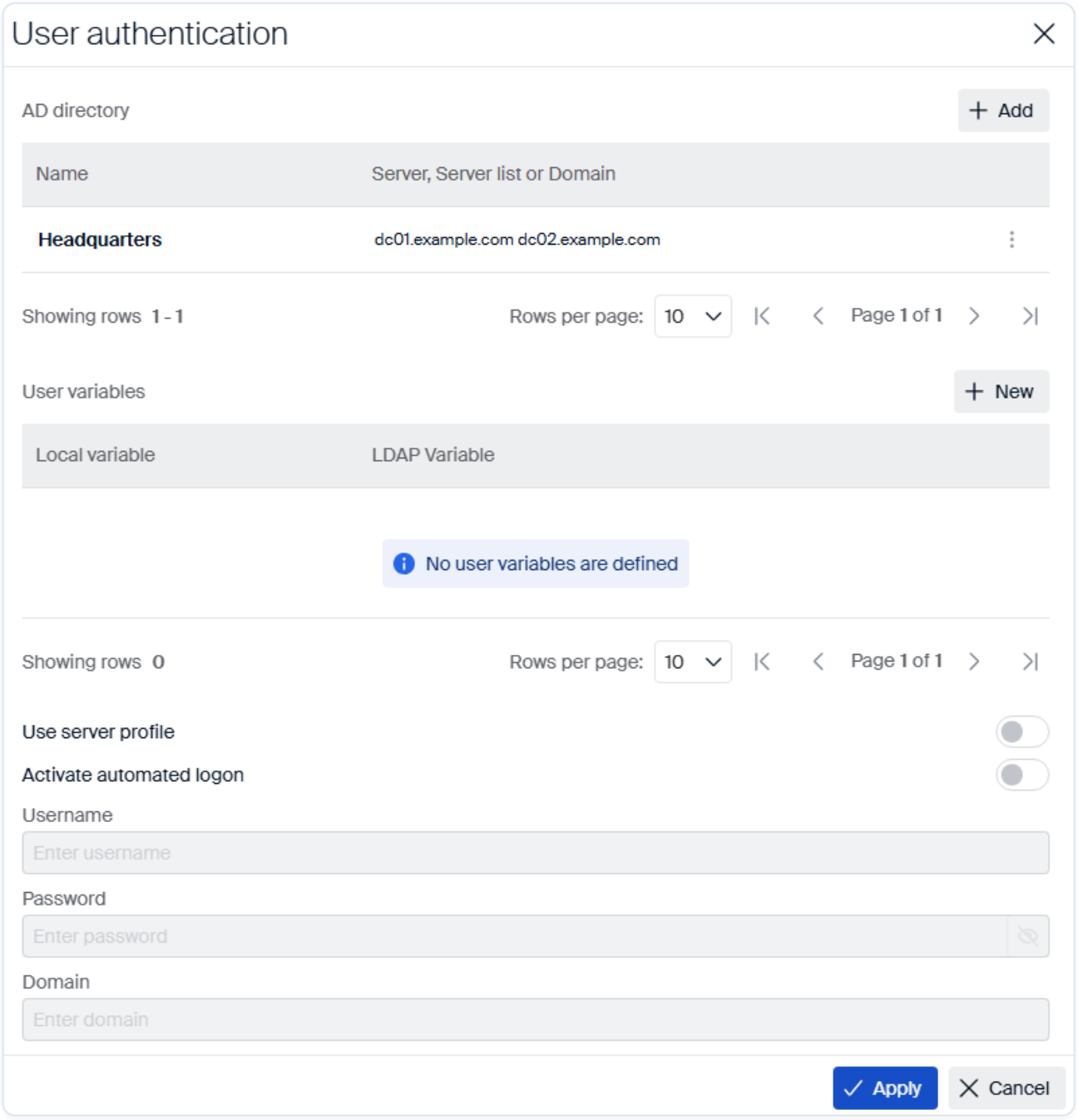
You can set both settings in the base configuration, for an OU, for a device and in configuration labels. In a label, **Suppress browser prompts** and **Trusted browser origins** are separate settings of the **User authentication** section. A label sets only the ones that you add to it. If a label sets an empty **Trusted browser origins**, devices get no list, even if their device configuration has one.

After the upgrade to UniconOS Management 2611, **Suppress browser prompts** is off and **Trusted browser origins** is empty in every configuration. If you turned on prompt suppression with advanced file entries before, these entries win over the two settings, because Scout sends them

afterwards. To manage prompt suppression in the **User authentication** section instead, set the two settings first. Then delete the advanced file entries for `SuppressPrompts` and `BrowserTrustedUris`.

ACTIVE DIRECTORY

With **Active Directory**, users sign in with their domain account and password. The devices need the packages **User authentication modules** and **Security libraries** to sign in to different domains.



In the settings dialog, define the domains in the **AD directory** table:

1. Click **Add**. A new row appears.

2. Enter a **Name** for the domain and click the check mark next to the field. Users see this name in the domain list of the sign-in dialog.

   The name can contain only the letters A–Z and a–z and the digits 0–9, for example `corp`. UniconOS Management Board refuses other characters in a new or changed name. Devices can't use a name with a dot, for example `corp.local`: they drop the settings of this domain, and users can't choose it when they sign in. Rename older entries with a dot.

3. Enter the **Server, Server list or Domain** and click the check mark next to the field. Enter the name or IP address of the domain controller. Separate several servers with spaces. If the server is not in the same subnet as the device, enter its fully qualified domain name (FQDN).

If you define more than one domain, users choose a domain from a list. To change a domain, open the menu at the end of its row. The menu lets you edit the entry, set its search base, make it the default domain or delete it:

MENU ITEM	DESCRIPTION
Edit	Changes the name or the server of the entry.
Change search base...	Opens a dialog where you enter the search base for this domain entry, for example `DC=int,DC=sampletec-01,DC=com`.
Set as default domain	The device preselects this domain on its sign-in and unlock screens. A star marks the default domain. The menu item changes to **Remove default domain**. Users can still choose another domain if the domain field is editable. Requires UniconOS 2609 or later.
Delete	Removes the entry. If it was the default domain, the default is cleared.

> **Note:**

> We recommend a Windows time server. If the system time of the domain controller and the device differ, Active Directory queries fail.

> The dialog also has settings for user variables, the server profile and automatic sign-in:

SETTING	TYPE	DESCRIPTION
User variables	Table	Local variables that are based on LDAP attributes. Click **New**, and then enter the **Local variable** and the **LDAP Variable**. You can use the variables in the device configuration and in application definitions. For the rules, see User variables.
Use server profile	Toggle	The device saves the client data on a server (server profile).
Activate automated logon	Toggle	Signs the device in automatically with the **Username**, **Password** and **Domain** below. For example, a kiosk can run under an Active Directory service account. The fields are disabled until you turn the toggle on.

ACTIVE DIRECTORY + SMARTCARD

With **Active Directory + Smartcard**, users sign in with a smart card. The devices need the middleware for the smart card, for example **Cryptovision sc/interface**, and the hardware drivers, for example **PCSC Lite**.

The dialog has the **AD directory** table, the **User variables** table, **Use server profile**, and **Activate automated logon** as for Active Directory. **Activate automated logon** appears only while **Allow logon with username + password** is selected. These settings are specific to smart cards:

SETTING	TYPE	DESCRIPTION
On Smartcard removal	Dropdown	What the device does when the user removes the card: **No action**, **Logoff current user** or **Lock screen**.

SETTING	TYPE	DESCRIPTION
Certificates Upload	Button	Click **Select files...** to upload one or more root certificates. The device verifies the certificate of the user against them. The area below the button lists the uploaded certificates, or shows **No certificates found**.
Allow logon with username + password	Checkbox	Users can also sign in with a user name and password, through the **Username & Password** link in the smart card dialog.
Show username + password dialog by default	Checkbox	The sign-in dialog opens with user name and password instead of the smart card prompt. Available while **Allow logon with username + password** is selected.

EVIDIAN

With **Evidian**, users sign in through Evidian identity and access management with a smart card or an RFID badge. The devices need the eLux® package **Evidian**.

In the settings dialog, define the Evidian servers in the **Evidian Server** table. Click **Add**, and then enter a **Name**, with letters and digits only, and the **Evidian Server or Server list**. Separate several servers with spaces. Use one of these formats:

```
http://<FQDN or IP address>:9764/soap

https://<FQDN or IP address>:9765/soap
```

The dialog also has settings for the sign-in method and for the shared secret of the Evidian server:

SETTING	TYPE	DESCRIPTION
Use Smartcard	Checkbox	Users sign in with a smart card.
Allow logon with username + password	Checkbox	Users can also sign in with a user name and password.
Secret	Text field	The shared secret of the Evidian server. Copy the value of the `ExternalRoamingSessionSecret` key from the registry of the Enterprise Access Management server. Do not encrypt it.

CITRIX

Citrix sets up a Citrix desktop lock: the device opens one published desktop or app of a Citrix store. In the **Citrix user authentication** dialog, enter the store and the resource that the device opens:

SETTING	TYPE	DESCRIPTION
URL	Text field	Address of the Citrix store.
Resource	Text field	The published desktop or app that the device opens.

To open a Citrix desktop after users sign in with another sign-in type, use the start scenario **Boot to VDI**, see Start scenario.

OPENID CONNECT

With **OpenID Connect**, users sign in through an identity provider. In the **OpenID Configuration** dialog, enter the data of your identity provider:

SETTING	TYPE	DESCRIPTION
OpenID authorise endpoint URL	Text field	Address of the authorization endpoint of the identity provider.

SETTING	TYPE	DESCRIPTION
OpenID redirect URL	Text field	Address that the identity provider redirects to after sign-in.
OpenID client ID	Text field	Client ID of the device configuration at the identity provider.
OpenID client secret	Text field	Client secret for the client ID.

SAML

With **SAML**, users sign in through a SAML identity provider. In the **SAML Configuration** dialog, enter the data of your identity provider:

SETTING	TYPE	DESCRIPTION
SSO URL	Text field	Single sign-on URL of the identity provider.
Issuer	Text field	Issuer of the identity provider (entity ID).
Audience	Text field	Audience restriction.
Certificate path on device	Text field	Full path of the certificate file on the device, including the file name, for example `/setup/elux/saml/cert.pem`.
IdP Certificate	Button	Click **Select file...** to upload the certificate of the identity provider. The area below the button shows **No certificate uploaded** until you select a file.

A configuration label applies its SAML settings only if it also contains the IdP certificate. Without the certificate, the label editor asks you to upload it.

> **Important:**

> The release notes list OpenID Connect and SAML as sign-in types that need additional setup with advanced file entries. See Known issues.

IMPRIVATA

With **Imprivata**, users sign in with an Imprivata badge or card. In the **Imprivata user authentication** dialog, enter the address of the appliance, the timeout and the default domain:

SETTING	TYPE	DESCRIPTION
Bootstrap URL	Text field	Address of the Imprivata appliance.
Authentication timeout	Number field	Time in seconds that the device waits for a password or PIN after the user taps a card. After this time, the device returns to the sign-in screen. The initial value is `60`.
Default login domain	Text field	Domain that the device preselects on its sign-in and unlock screens. Enter up to 255 characters. The name must match a domain that the device knows. Case does not matter. Leave the field empty to remove the default. Requires UniconOS 2609 or later.

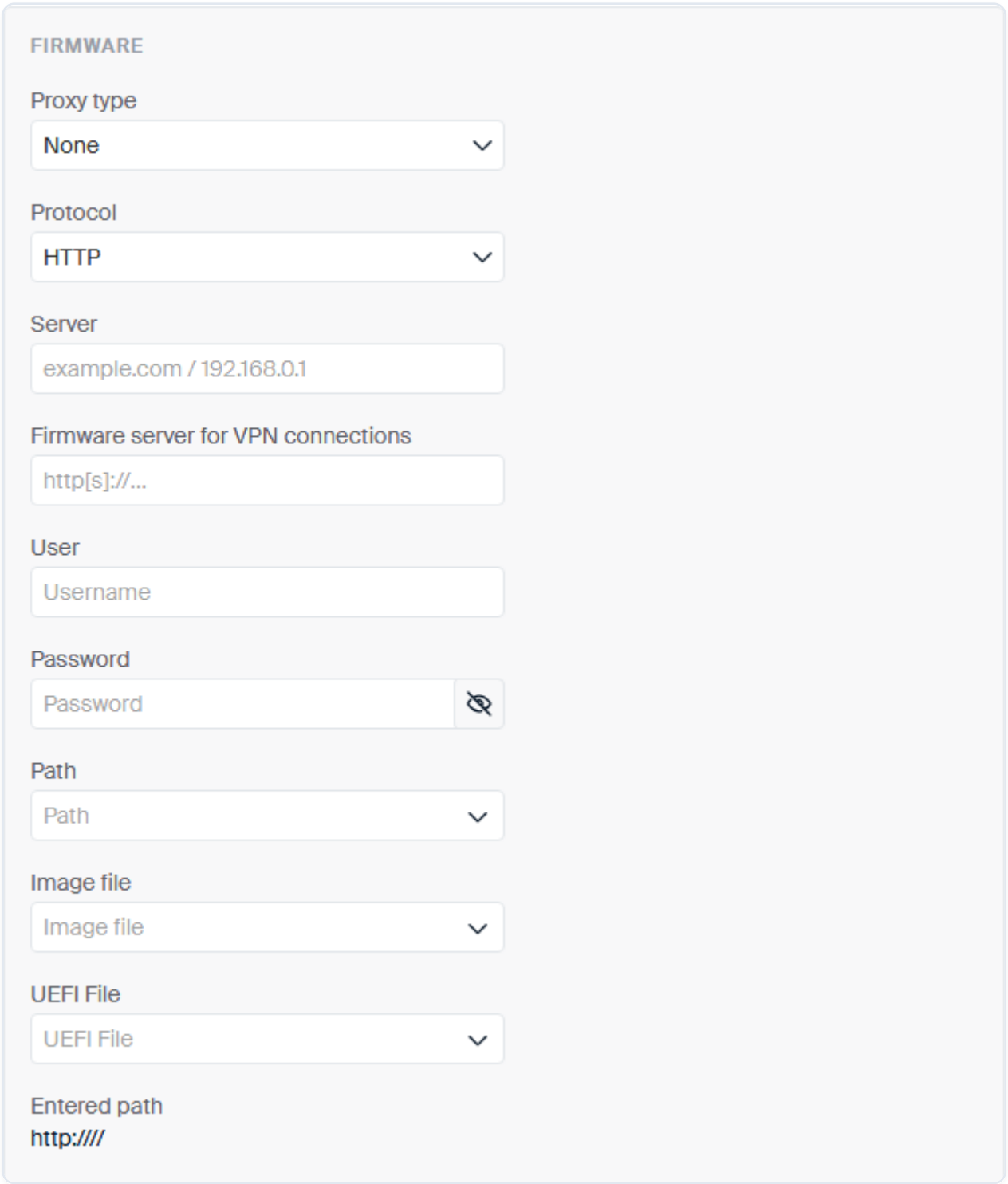
For the full setup, see Scout in the Imprivata guide.

ENTRA ID

Entra ID has no settings dialog. The release notes state that Entra ID is currently supported only for Azure Virtual Desktop use cases. See Known issues.

Device configuration / Firmware

In the **Firmware** section, you define where your devices get their firmware and how they check it. You also decide how users can defer updates and what the device shows before an update runs. You find **Firmware** in the list of sections on the left of the device configuration. For how these settings look on the device, see Firmware.



FIRMWARE SERVER

The fields **Protocol**, **Server**, **Path** and **Image file** build the address that devices use for a firmware update. The line **Entered path** below **UEFI File** shows this address.

With these settings, you tell the devices where they get their firmware and how they download it:

SETTING	TYPE	DESCRIPTION
Proxy type	Dropdown	Decides how the device gets its software packages: **None**, **Static (Consumer)**, **Static (Provider)** or **Dynamic**. See Proxy type.
Provider	Text field	Appears when **Proxy type** is **Static (Consumer)**. Enter the address and port of the proxy device, for example `192.168.0.1:3128`.
Protocol	Dropdown	The network protocol that devices use to transfer software packages from the web server: **FTP**, **FTPS**, **HTTP** or **HTTPS**.
Server	Text field	The name (FQDN) or IP address of the web server that holds the UniconOS software packages and the image definition file.
Firmware server for VPN connections	Text field	Optional. The address of an alternative web server for devices that are connected through VPN. Start the address with `http://` or `https://`.
User	Text field	Optional. The user name for the web or FTP server.

SETTING	TYPE	DESCRIPTION
Password	Text field	Optional. The password for the user name in **User**.
Path	Dropdown	The directory path of the UniconOS software packages on the web server. Separate directories with slashes. For a container from **Software**, enter the path of its folder on your web server. The **Firmware settings** of an image in **Software** show this path, see Configuring the devices.
Image file	Dropdown	The name of the image definition file (IDF) on the web server. The IDF defines the software packages that devices install. You create it in UniconOS Management Board under **Software** > **Images**, see Creating an image. The name can contain only letters, digits and `. _ + ~ -`. It must start with a letter, a digit or `_`, and it can have up to 127 characters. UniconOS Management Board refuses a new or changed name that breaks these rules.
UEFI File	Dropdown	Optional. The UEFI file in the container. It defines which UEFI firmware is installed on which device type.

In the **Path**, **Image file** and **UEFI File** dropdowns, select a value from the list or enter your own.

Proxy type

A proxy device downloads software packages and passes them on to other devices. This helps devices on narrow-band connections. In **Proxy type**, you choose whether and how the device takes part in proxy updates:

- **None**: the device downloads the software packages from the web server and takes no part in proxy updates.
- **Static (Consumer)**: the device downloads the software packages through a proxy device that you set up. Enter the address of the proxy device in **Provider**. For the setup, see Static proxy.
- **Static (Provider)**: the device is the proxy device for other devices.
- **Dynamic**: the devices of a subnet update through one of them. The system automatically selects a device with an update partition as the provider, and the other devices of the subnet are consumers. If you use HTTP, leave **User** and **Password** empty. See Dynamic proxy.

SECURITY

Before an update, the device can check signatures. The update runs only if the signatures are valid and verifiable. The device records the results in the update log file. Choose which signatures the device checks:

SETTING	TYPE	DESCRIPTION
Image definition files (.idf)	Checkbox	The device checks the signature of the image definition file before an update.
UniconOS software packages	Checkbox	The device checks the signatures of the software packages before an update. This check needs an update partition on the device. A device without an update partition can check only image definition files.

To verify an IDF signature, the device needs the root certificate and the signature certificate in `/setup/cacerts`. For more information, see Firmware security through signatures.

Both check boxes are cleared by default. Devices then install the image and its packages without checking any signature, also if the files were changed on the server. To make devices refuse

unsigned or changed files, select both check boxes and sign your images, see Checking signatures.

REMINDER

With the reminder settings, you decide whether and how users can defer an update. Users see the deferral options in the message that informs them before an update. This message appears when you run the **Update software** command with **Inform users**. It also appears when the device starts an update itself with user confirmation.

Set how often users can defer an update and which delays they can choose:

SETTING	TYPE	DESCRIPTION
Number of allowed deferments	Dropdown	How often users can defer an update, from `0` to `3`. With `0`, users can't defer.
Delays until next reminder	Dropdown	The time intervals that users can choose for the next reminder: **15 minutes**, **30 minutes**, **1 hour**, **2 hours**, **4 hours**, **8 hours** or **Deferment until reboot**. You can select several.

The message that users see is described in User information before update.

UPDATES THAT THE DEVICE STARTS ITSELF

The device can check on its own whether a firmware update is necessary. An update is necessary if the image definition file on the web server differs from the local version. The device then downloads the missing software packages and installs them. Because the device starts the update, it uses the firmware settings that are stored locally on it.

Choose when the device checks for a new image version:

SETTING	TYPE	DESCRIPTION
Check for new image version on start	Toggle	The device checks at each start whether an update is necessary.

SETTING	TYPE	DESCRIPTION
Check for new image version on shutdown	Toggle	The device checks at each shutdown whether an update is necessary.

If the device can set up its VPN connection only after the desktop has loaded, it does not update at start. Users can continue to work. Devices that connect through the Scout Cloud Gateway are not affected.

When you turn on **Check for new image version on start**, the group **User interaction and formatting** appears. Its settings decide what users can do before the update and whether the device formats the system partition first:

SETTING	TYPE	DESCRIPTION
Inform user for	Number field	How many seconds the message before the update stays on the screen. With `0`, the message stays until the user clicks a button.
User can cancel command	Checkbox	Lets users cancel the update for good. The device then doesn't restart the update by itself.
Format system partition before update	Checkbox	The device formats the system partition before it runs the update.

UPDATE NOTIFICATIONS AND BATTERY LIMITS

In this group, you decide for each firmware operation whether the device shows notifications and which minimum battery level it needs. The firmware operations are **Deliver software**, **Update software** and **UEFI update**. Each one has a pair of settings, a notification switch and a minimum battery level:

- **Show notification on delivery / on update / on UEFI update**: on by default. Turning a switch off makes that operation run silently on the device.

    - From UniconOS 7.2603.1000 and UniconOS 7.2605.0, turning a switch off hides the low-battery messages for that operation.
    - From UniconOS 7.2607.0, turning a switch off also hides the notice and prompt shown before that operation, even when **Inform users** is selected for the command. The update switch also overrides **Inform user for** and **User can cancel command** for updates that the device checks for on its own. Users can then no longer defer or cancel the operation.

- **Minimum battery level for delivery / for update / for UEFI update (%)**: a whole number from 0 to 100. The default is 50. Setting the value to 0 turns the check off.

  The battery check applies only when the device runs on battery power. A device with no battery, or one that is connected to power, always goes ahead. On battery, the operation starts only at or above this level. A device below the level does not run the operation, and its entry in **Command History** shows **Failed: Low battery**.

Each switch and its battery level are independent: turning a notification off does not turn off the battery check.

Devices on a version that does not know these settings ignore them and keep their built-in behavior. The settings follow the normal inheritance from the base configuration to OUs and devices. You can also set them in a configuration label, see Dynamic device configuration.

Device configuration / Desktop

The **Desktop** section of the device configuration controls the device's language, background, time settings, and the desktop information overlay.

LANGUAGE

In the **Desktop** section, **Language** sets the system language of the devices. The keyboard layout is a separate setting, see Keyboard and Mouse. For how the language setting looks on the device, see Desktop.

The list shows the languages sorted by name: **Croatian**, **Czech**, **Danish**, **Dutch**, **Dutch (Belgium)**, **English (UK)**, **English (US)**, **Estonian**, **Finnish**, **French**, **French (Belgium)**, **French (Switzerland)**, **German**, **German (Switzerland)**, **Greek**, **Hebrew**, **Hungarian**, **Icelandic**, **Italian**, **Japanese**, **Korean**, **Norwegian**, **Polish**, **Portuguese**, **Portuguese (Brazil)**, **Romania** (Romanian), **Russian**, **Slovak**, **Slovenian**, **Spanish**, **Swedish** and **Turkish**.

Hebrew and **Korean** are new in UniconOS Management 2611, in the device configuration and in the **Desktop** section of configuration labels. Devices need UniconOS 2607 or later for these two languages.

DESKTOP INFORMATION OVERLAY

The desktop information overlay shows a small panel on the eLux desktop with device details, useful for identifying a device at a glance. The overlay itself is set on the device, but with UniconOS Management you configure it centrally in UniconOS Management Board instead.

- **Show desktop info overlay**: off by default. Turning it on reveals **Theme** and the **Desktop overlay information** panel.
- **Theme**: **Light** (default) or **Dark**.
- Changes apply after the device restarts.

The **Desktop overlay information** panel lists the fields the overlay shows, grouped into four fixed sections:

SECTION	FIELDS
System	Device name, Domain, Operating system, Boot time
Network	IPv4 address, MAC address
User	Logged in user
Scout server	Scout server, Manager address

- All nine fields are selected by default. Click the **X** on a field to remove it, or **+** on a section to add back a field from that section. The trash icon removes a whole section with its fields. **Add section** adds a removed section back; it is unavailable once all four are shown.
- If you remove every field, the device shows no overlay.
- **Preview** shows the panel with sample values, not the device's real data.
- The overlay has a fixed position on the desktop and no custom section names: entries can only be added to or removed from these four sections.

Scout sends the overlay settings to every managed device, including when the switch is off. A value set directly on the device, for example through an advanced file entry, is overridden by the UniconOS Management Board settings at the next configuration update.

Requires UniconOS 2607 (eLux 7.2607) or later on the device.

Device configuration / Printer

The **Printer** section defines the printers that devices set up and the print services that they offer. You find **Printer** in the list of sections on the left of the device configuration. The **Defined printer** table lists the printers. Below the table, **LPD print service** and **TCP direct-print** let devices accept print jobs from other computers, and **ThinPrint** selects the channel for ThinPrint. For the printer settings on the device, see Printer.

ADDING A PRINTER

In edit mode, click **Add printer** above the **Defined printer** table. To change or delete a printer, open the menu at the end of its row and click **Edit** or **Delete**. The **Add printer** dialog has these settings:

SETTING	TYPE	DESCRIPTION
Name	Text field	The name under which applications on the device show the printer. Special characters such as `-`, `.`, `/` and `#` aren't allowed.
Connection type	Dropdown	How the printer is connected to the device: **Network**, **Parallel**, **Serial (COM1)** to **Serial (COM4)**, or **USB**.
Text filter	Toggle	Lets the device print plain text on the printer, for example from a local shell.
Device driver	Dropdown	The driver that the device uses for the printer. **None** sends the print data unchanged. **PCL2** prints in PCL format on printers that don't understand PostScript. **Other driver** uses the CUPS driver in **Driver name on the device**.

SETTING	TYPE	DESCRIPTION
Driver name on the device	Text field	Appears for **Other driver**. The name of the CUPS driver without `.ppd`, for example `zebra` for a Zebra label printer. The device then uses `zebra.ppd`. Up to 251 characters: letters, digits and `. _ + : / -`.
Bitrate	Dropdown	Appears for a serial printer. The transmission speed of the printer, from `1200` to `115200`. **<System>** uses the speed of the serial port of the device.
Printer address	Text field	Appears for **Network**. The IP address or host name of the printer or print server.
Printer queue	Text field	Appears for **Network**. The name of the print queue, for example the share name of the printer.

Below these settings, the dialog has three groups:

- **Server-side**: In **Driver name**, enter the name of the printer driver on the server for printing from a Windows session, for example with Citrix® auto-created printers. Spell it exactly like the driver installed on the server, with the same uppercase and lowercase letters and spaces.
- **ThinPrint**: Turn on **Connect** to print through ThinPrint. In **Class**, you can enter a ThinPrint class.
- **CUPS (eLux® 7.2601 or later)**: The PPD model and the printer options, see CUPS settings.

Click **Confirm** to add the printer to the table. A driver name that was set in an earlier version appears as **Other driver**, and UniconOS Management Board keeps it when you save the printer.

Devices with UniconOS use the bitrate that you select for a serial printer. Select the bitrate of the printer rather than **<System>**. With **<System>**, a device with UniconOS 2611 or earlier can stop

while it sets up its printers at startup. It then appears offline in UniconOS Management Board until it is switched off and on again.

CUPS SETTINGS

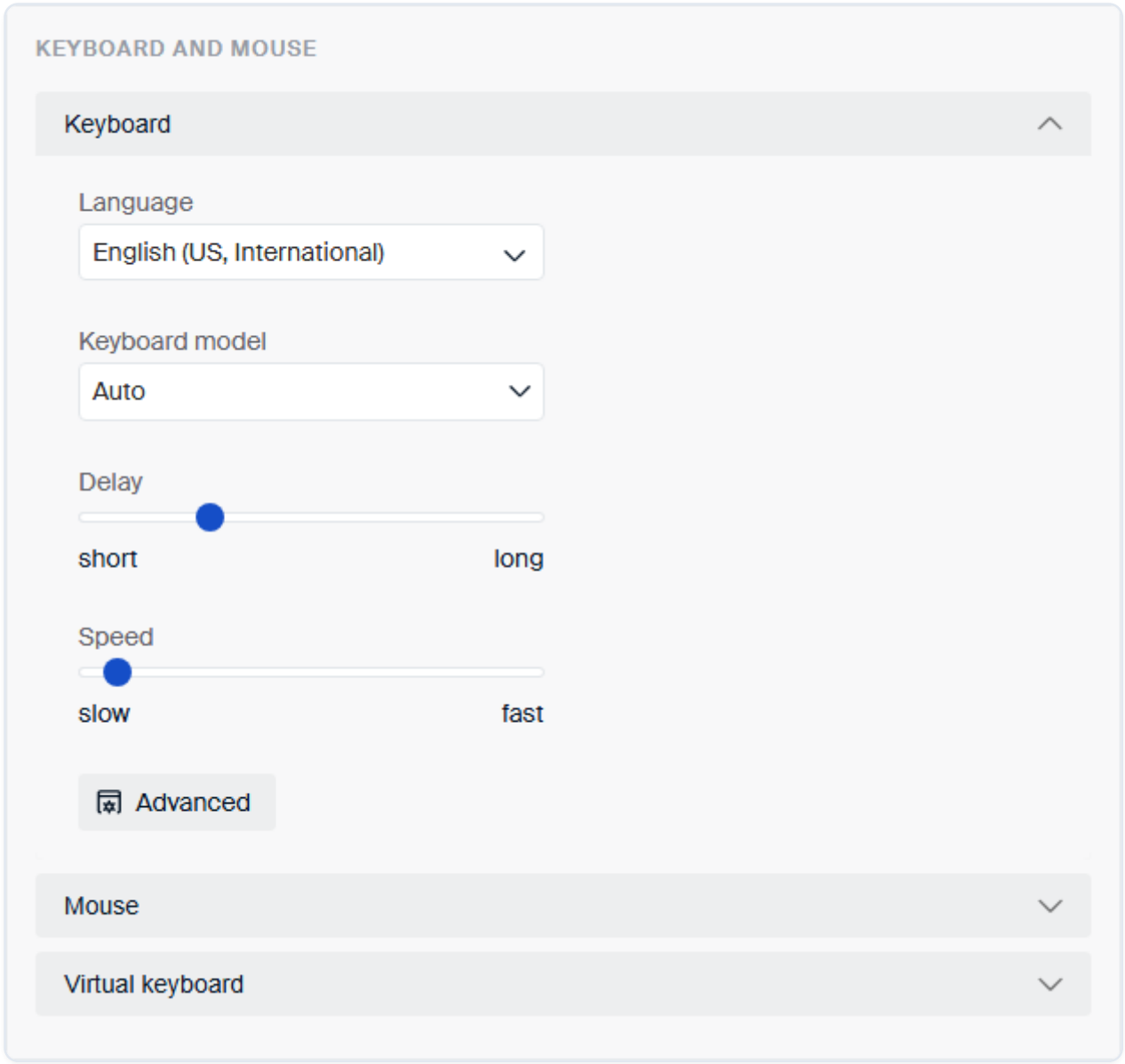
In the **CUPS (eLux 7.2601 or later)** group, you set how devices with UniconOS 2601 or later set up the printer in CUPS, for example the label size of a label printer:

SETTING	TYPE	DESCRIPTION
PPD model	Text field	The PPD file that the device uses instead of the device driver, for example `raw`. The device adds `.ppd` unless the name already contains it or is `raw`. Up to 255 characters, `.ppd` included: letters, digits and `. _ + : / -`.
Printer options	Text field	CUPS options of the printer as `key=value` or `key`, separated by commas, for example `PageSize=A4,Resolution=300dpi`. Keys and values can contain only letters, digits and `. _ -`. Up to 1,024 characters.

If a value breaks these rules, the text below the field turns red, and you can't confirm the dialog. The device would ignore such a value, and one wrong option makes it ignore all options of the printer. Devices with eLux RP 6 ignore both settings.

Device configuration / Keyboard and Mouse

In the **Keyboard and Mouse** section, you define the keyboard layout and how the keyboard behaves. You also set up the mouse, the touchpad and the virtual (on-screen) keyboard. You find **Keyboard and Mouse** in the list of sections on the left of the device configuration. For how these settings look on the device, see Keyboard and Mouse.



KEYBOARD

These settings define the keyboard layout and model, and how a key repeats while a user holds it down:

SETTING	TYPE	DESCRIPTION
Language	Dropdown	The keyboard layout. From UniconOS Management 2611, the list also offers **German (Neo 2)**.
Keyboard model	Dropdown	**Auto**: the system identifies the keyboard type by itself. Alternatively, choose **Cherry G81-8000 keyboard**, **Desko**

SETTING	TYPE	DESCRIPTION
		MCx 4717 keyboard, **MFII keyboard**, **Sun Type6 keyboard**, **Trimo 3270 keyboard** or **Trimo 97801 keyboard**.
Delay	Slider	How long a user must hold a key until the letter repeats. The scale runs from short to long, from `100` to `1510`.
Speed	Slider	How fast a letter repeats while a user holds a key. The scale runs from slow to fast, from `10` to `250`.
Advanced	Button	Opens the dialog for further keyboard settings.

Click **Advanced** to open the **Advanced keyboard settings** dialog. Its settings control special keys, such as dead keys and the numeric keypad:

SETTING	TYPE	DESCRIPTION
Dead Keys	Toggle	A dead key produces visible output only after a second keystroke. Accent keys are dead keys, because users press them before the character key. Some hardware platforms and applications don't support dead keys.
Console switch	Toggle	Lets users switch between consoles with key combinations.

SETTING	TYPE	DESCRIPTION
Extended keys	Toggle	Enables multimedia keys and other keys with special functions.
Handle Caps Lock always like Shift	Toggle	With Caps Lock active, the number keys above the letter keys output special characters instead of numbers, as they do with the Shift key.
Num Lock	Dropdown	**On**: the numeric keypad is enabled when the device starts. **Off**: the numeric keypad is disabled when the device starts. **Auto**: the numeric keypad is enabled on mobile devices and disabled on other devices.

MOUSE

These settings define how the mouse and the touchpad behave:

SETTING	TYPE	DESCRIPTION
Show mouse pointer	Toggle	Shows the mouse pointer. Turn it off to hide the pointer, for example on an information board.
Touchpad	Dropdown	For devices with a touchpad. **On**: the touchpad is enabled. **Off**: the touchpad is disabled. **Auto**: the touchpad is disabled while a mouse is plugged in.
Double-click speed	Slider	The time interval between the two clicks of a double-click.

SETTING	TYPE	DESCRIPTION
		The scale runs from slow to fast, from `100` to `900`.
Pointer speed	Slider	How fast the mouse pointer moves. The scale runs from slow to fast, from `1` to `100`.
Left-handed	Toggle	Swaps the primary and the secondary mouse button.

VIRTUAL KEYBOARD

The virtual keyboard is an on-screen keyboard that device users can operate with a mouse, a touchscreen or another pointing device, without a physical keyboard.

In the **Virtual keyboard** panel, you decide whether the keyboard starts by itself, how it looks and whether device users can close it:

SETTING	TYPE	DESCRIPTION
Start virtual keyboard automatically	Toggle	Off by default. When on, the keyboard shows at startup, on the logon screen, the lock screen, and the desktop.
Theme	Dropdown	**Dark** (default) or **Light**.
Layout	Dropdown	**Compact** (default) or **Full keyboard**.
Prevent closing the virtual keyboard	Toggle	Off by default. When on, the keyboard has no close button on the logon screen, the lock screen and the desktop, so that device users can't close it by accident.

Theme, **Layout** and **Prevent closing the virtual keyboard** apply whether or not the keyboard starts automatically. Turning off **Start virtual keyboard automatically** does not turn the keyboard

off. Device users can still show it themselves from **Quick System Access** > **Accessibility**, unless the matching user right is locked.

The virtual keyboard requires UniconOS 2609 or later, with the **onboard** package in the device image. On earlier versions, Scout saves the settings without an error, but the device ignores them. Configure the keyboard with the file-entry method described in On-Screen Keyboard instead. A file entry for the same key overrides the panel value. To switch a configuration to the panel, set the values in the panel first, and then delete the `[VirtualKeyboard]` file entries.

Preventing users from closing the keyboard

On a touch-only device without a physical keyboard, the virtual keyboard is the only way to type. Turn on **Prevent closing the virtual keyboard** for such devices, so that a stray tap can't close the keyboard. To have the keyboard come up by itself when the device starts, also turn on **Start virtual keyboard automatically**.

The setting removes only the keyboard's own close button. Device users can still hide the keyboard on purpose, with the same switches that show it: in **Quick System Access** > **Accessibility**, also on the logon and lock screens, and with **Show on screen keyboard** in the device configuration on the device. Locking the user right **Enable virtual keyboard** removes these switches.

On the device, the setting is called **Hide keyboard close button** and is in the **Accessibility** section of the device configuration, where device users can turn it on and off. Scout sends your value to the devices with every configuration update, also while the setting is off. By default, the device then replaces a value that a user changed on it. To stop device users from changing the setting at all, lock its user right, see User rights.

The setting needs UniconOS 2611 or later on the device, with the **onboard** package of the same release in the image. Earlier UniconOS versions ignore it, and their keyboard keeps its close button.

User rights

Four user rights let you stop device users from showing the keyboard themselves, from changing its theme and layout, and from changing **Hide keyboard close button** on the device: **Enable virtual keyboard**, **Virtual keyboard theme**, **Virtual keyboard layout** and **Prevent closing the virtual keyboard**. To find them, open the **Security** section of the device configuration, go to **User rights**, and look under **Device configuration** > **Mouse/Keyboard** in the tree. See User rights for how locking works, and for a caution about devices with no physical keyboard.

If **Retain local device configuration (unlocked fields)** is turned on under **Preferences** > **Devices**, devices keep the values that users set in unlocked fields. In this case, also lock **Prevent closing the virtual keyboard**, so that your value reaches the devices.

Upgrading to UniconOS Management 2611

After the upgrade, **Prevent closing the virtual keyboard** is off in all device configurations, and Scout sends this value to the devices. On a device where someone turned on **Hide keyboard close button**, the first configuration update after the upgrade can turn it off again, and the close button comes back. Configurations that lock **Mouse/Keyboard**, directly or by locking **Device configuration** as a whole, also lock the new entry. If devices need the setting, turn it on in UniconOS Management Board right after you upgrade Scout.

Upgrading to UniconOS Management 2609

After the upgrade, device configurations (base, OU or device) that lock **Mouse/Keyboard**, directly or by locking **Device configuration** as a whole, also lock the three new entries. On devices that run UniconOS 2609, device users then lose the keyboard toggle and can't change the theme or layout. By default, the keyboard also doesn't start automatically. On a device with no physical keyboard, this leaves users no way to bring up the on-screen keyboard, not even on the logon screen. Review these configurations right after you upgrade Scout, whichever UniconOS version the devices run.

Configuration labels are the exception. A label saved before 15.2609 that locks **Mouse/Keyboard** does not lock the three new entries on the devices it covers, even though the label editor shows them as locked. If you rely on a label to lock keyboard settings, lock the three entries in it explicitly.

Other ways to configure it

- **Configuration labels**: the same four settings are available under **Device management** > **Configuration labels** > label > **Keyboard and Mouse**.
- **REST API**: the same four settings are also available through the REST API. Changing them needs the same **Mouse/Keyboard advanced** administrator right as the other advanced keyboard settings. For the full reference and how scripts sign in, see Using the REST API.

Commands and schedules

Commands can be issued to OUs, Dynamic Device Groups and individual devices.The following commands are available in the Technical Preview:

- Switch on
- Restart
- Switch off
- Send message

- Reset configuration
- Deliver software (from Scout 15 2304)
- Pre-defined command (from Scout 15 2304)

If a command is not executed immediately, it is saved as a schedule. A schedule can be defined for one-time execution or for periodic execution.

Schedules are displayed for each OU in the device list on a separate tab. The number of existing schedules is additionally shown in the for each OU, along with the number of existing devices and applications.

You may create commands or schedules for the following target devices:

- a device of a device list (all three views)
- the devices of an OU in the OU structure, optionally additionally on all devices in subordinate OUs
- the devices of a Dynamic Device Group

To create a command or schedule, open the **Actions** menu for the target devices. To do this, click on ⋮ to the right of the device, OU or Dynamic Device Group.

Creating a command

> **Note:**
>
> Means a Scout command that is executed immediately.

1. For the target devices, open the **Actions** menu, then choose **Commands**, and then the desired command.
2. Leave the setting for **Type** (time) on **Now**.

   *If you do not want to execute a command right away, set a one-time execution time or even a periodic execution, see **Creating a schedule**.*
3. In the **Command** dialog, edit the remaining fields.Common options are the following:

OPTION	DESCRIPTION
Inform users	The relevant users will receive a message before the command is executed.
Include subordinated OUs	The command is additionally applied to all devices in subordinate OUs.

For further information, see Command options in the Scout guide.

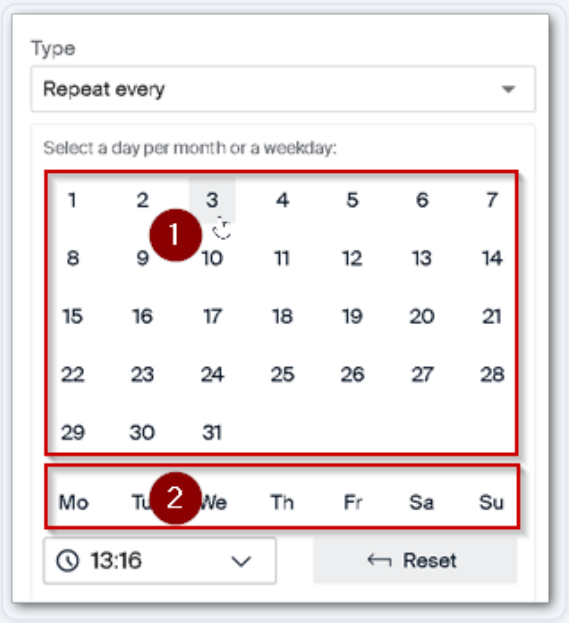
4. Click **Execute**.

The command is executed immediately.

Creating a schedule

> **Note:**
>
> Means a Scout command that is scheduled for later execution.

1. For the target devices, open the **Actions** menu, then choose **Commands**, and then the desired command.
2. Under **Type** (time), select either `Once` or `Repeat every`.
3. If you intend to plan a one-time execution, then select the date and time.
4. If you intend to plan a periodic execution, select one day per month one day of the week below it.

	
1	Option A: Choose a fixed day per month (for example, every 5th.)
2	Option B: Choose a fixed day per week (for example, every Tuesday)

Specify the time.

5. In the **Command** dialog, edit the remaining fields. For further information, see Command options in the **Scout** guide.
6. Click **Execute**.

A schedule is created for the planned one-time or periodic execution of the command.

Help-desk and remote maintenance

Helpdesk functions such as mirroring a device are available to assist users with problems.

You can refer to the following:

- Mirroring a device
- Retrieving diagnostic data

Mirroring a device

-- from eLux® RP 6 2302 --

Individual devices can be mirrored if they are switched on and the following requirements are met.

REQUIRES
The target device must have installed **eLux RP 6 2209** or later.
A VNC server must be installed on the target device. For this, the VNC Server extension feature package included in the eLux package **XOrg** must be installed.
Mirroring must be enabled and configured for the target device in the **Mirroring** section of the device configuration.
Encrypt transmission must be off in the **Mirroring** section. UniconOS Management Board encrypts its mirror sessions itself, and it can't mirror a device while this option is on.

1. Select a device on a device list (all three views).
2. Click ⋮ to open the device's **Actions** menu and choose **Mirror**.

   > **Note:**
   >
   > Your browser must allow pop-ups.
   >
   > The mirrored device is displayed in a new browser tab.

3. If a mirroring password is set in the **Mirroring** section of the device configuration, enter the password on the new tab.

Depending on the device configuration, users must confirm the mirroring request within a defined time span before a session can be initiated.

To let the device show and log which administrator requested the session, turn on **Allow from UniconOS Management only** in the **Mirroring** section of the device configuration. Without it, the device records the requester as "unknown". While the setting is on, the device refuses VNC viewers that connect to it directly. On devices with UniconOS 2609 or later, the system message then shows the name of the administrator, together with the address `127.0.0.1`.

If UniconOS Management Board can't set up the session within about 30 seconds, the mirror tab shows **The mirror connection could not be set up in time. Check that the device is online and try again. If this keeps happening, ask your administrator to check the Scout server connections.** If several Scout Servers share one Scout database, check **Node health**, see Checking the health of the servers.

For further information, see Configuring mirroring in the Scout guide.

1	New browser tab for the mirror session
2	IP address of the mirrored device
3	Options for the mirror session
4	Permanent system message

Role of the system message

Throughout a mirror session, a system message (4) will be displayed on the screens of the user (mirrored) and administrator (mirroring):

- The message remains permanently in the foreground and indicates that mirroring is active.
- The mirrored user can end the session at any time (**Quit** button).
- The mirrored user can specify that the administrator may only read, which means that he is not allowed to take control of the mirrored device (**Read only** option).

Configuring a remote session

During a mirror session, there are several options available.

- At top right, click **Configure remote session**:

1	Disconnect mirror session. The session window and tab are closed.
2	Send `Ctrl+Alt+Del` into the session, for example to make a logon
3	Show session window in full screen mode. To return to window mode, press `Esc`
4	Fit session window to browser window.Depending on the aspect ratio, the entire screen of the mirrored device may not be displayed.

5	Adjust session window in maximum size to the local situation (default)

The set options for the session window size are saved in the browser for the current user.

Retrieving diagnostic data

> **Note:**
>
> We recommend that you temporarily switch on enhanced logging on the device before device diagnostics. This generates and makes available diagnostic files on a larger scale.

1. Select a device on a device list (all three views).
2. Click ⋮ to open the device's **Actions** menu and choose **Request diagnostic files**.

The scripts defined in the system template are executed. All log and configuration files are packed and made available for download in the browser. In the UniconOS Management Board, you will receive notifications about the progress and a link to the download.

To choose optional diagnostic templates for the request, you need the object right **Select diagnostic templates** on the device. Without this right, the device sends the default diagnostic files, see Existing rights UniconOS Management Board now enforces.

If you request the diagnostic files of a device again while they are still being collected, UniconOS Management Board doesn't start a second collection. It tells you that the files "were already requested and are still being prepared" and when they were requested. This also applies if another administrator requested them, and the files then reach each of you.

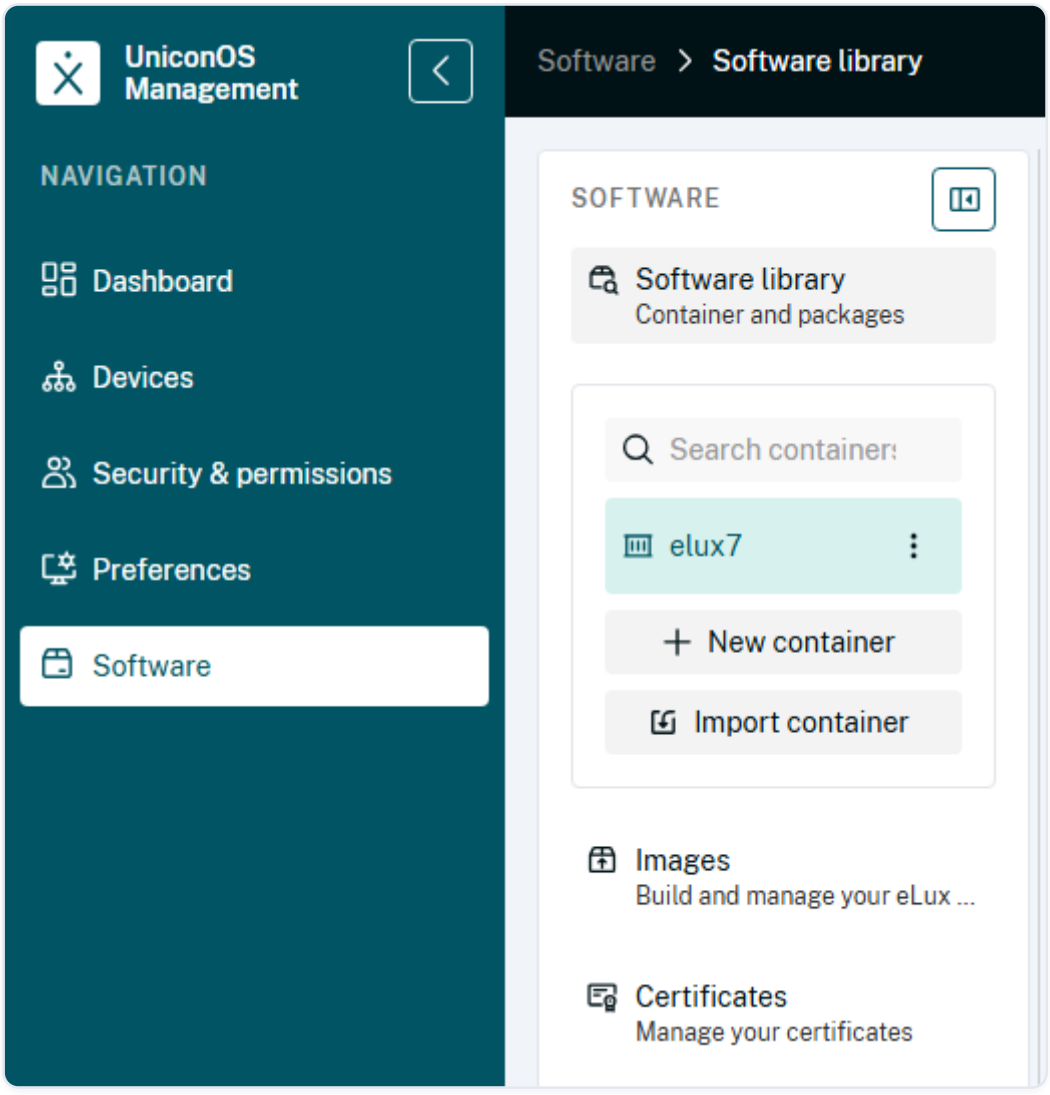
If the collection takes longer than usual, by default more than 5 minutes, UniconOS Management Board tells you so. The files still appear in your notifications when they arrive. If you request the files again after this time, UniconOS Management Board offers **Start a new collection anyway**. A request that delivers no files ends after 30 minutes.

For further information, see Device diagnostics in the **Scout** guide.

Software

In **Software**, you build the images for your UniconOS devices. An image lists the software packages that a device installs, from the operating system to the applications. You keep the packages in containers, put images together from them and sign the images, all in UniconOS Management Board. Software takes over the tasks of ELIAS, so you work with the same kinds of objects as before: containers, packages, images and certificates.

To open it, click **Software** in the **Navigation** menu on the left.



Next to the **Navigation** menu, the **Software** menu opens. Its three entries lead to the pages where you do the work:

ENTRY	WHAT YOU DO THERE
Software library	Create and import containers, add software packages to them, and export packages. You also choose which features of a package new

ENTRY	WHAT YOU DO THERE
	images get. Below this entry, the menu lists your containers.
Images	Build images from the packages of a container, let Scout add the packages that they depend on, and sign, export or delete the images.
Certificates	Manage the certificates of a container: the certificates that confirm the signatures of the packages, and the certificate that signs your images.

To give the pages more room, you can hide the **Software** menu, see Interface.

Your devices don't download their software from UniconOS Management Board, but from a web server or an FTP server that publishes the container folders. You set up this server yourself, see Publishing a container to devices.

Software appears in the **Navigation** menu only if both of these conditions are met:

- The components **Software Management** and **Software Management backend** are installed. A new installation of Scout includes both by default. An update from an earlier version adds them only if you select them, see Installing Software Management.
- You hold the rights **Software** and **View software**, see Software rights.

Your rights also decide which actions you can use in Software. An action that your rights don't allow stays visible but is greyed out. Pointing to it shows **You do not have the rights for this. Ask an administrator.**

Containers, packages and images

These are the objects that you work with in Software:

TERM	MEANING
Container	A folder that holds software packages and the images built from them. Each container has its own certificates.
Package	A software package in an `.epm` file, for example an application or a driver

TERM	MEANING
Feature	A part of a package in an `.fpm` file. An image can include a feature or leave it out, unless the feature is required.
Image	An image definition file (`.idf`). It lists the packages and features that a device installs.
Signature	A `.sig` file next to a package or an image file. A device can check it before it installs an update.

If you have worked with ELIAS 18, you find its libraries here:

ELIAS 18	SOFTWARE
Container Library	The container list in the **Software** menu, below **Software library**
Package Library	**Software library**, tab **Package overview**
Container Explorer	**Software library**, tab **Container explorer**
Image Library	**Images**
Certificate Library	**Certificates**
Access rights with the roles **Info**, **User** and **Admin**	The **Software** rights and the container roles of each administrator

To take a container over from ELIAS 18, export it in ELIAS 18, and import it with **Import container** in the **Software library**. Its packages and complete images come over, but not its templates, certificates and access rights, see Importing a container.

From packages to devices

To give your devices a new image, you usually take these steps:

1. In the **Software library**, create a container, or use the container that the installation created.
2. Add the software packages to the container, for example the AllPackages archive of a UniconOS version.

3. Under **Certificates**, check that the container trusts the signers of the packages. To sign images, add a signing certificate.
4. Under **Images**, create an image and select its packages.
5. Click **Solve dependencies**.
6. Click **Save image**.
7. If your devices check image signatures, sign the image.
8. Publish the container folder with a web server or an FTP server, for example with IIS, see Example of publishing the containers with IIS.
9. In the **Firmware** section of the device configuration, enter the values that **Firmware settings** shows for the image, see Configuring the devices.

The devices get the changed configuration when they restart, and they install the new image with their next firmware update.

Activity log and system status

Scout records the changes made in Software in the activity log, with the administrator and the object. You find the log under **Security & permissions** > **Manage administrators** > **Activity log**.

The entries of Software have types that start with **ELIAS:**. For example, creating a container writes an entry of the type **ELIAS: Add** with the object `Add container "<name>"`, and deleting it writes **ELIAS: Delete** with `Delete container "<name>"`. Signing an image and importing certificates write entries of the type **ELIAS: Sign**. Solving dependencies and exporting a package write no entry. To list only the entries of Software, click **Filter**, select **All ELIAS actions** under **Value**, and click **Apply**.

Entries of the type **ELIAS: Sign** are recorded at every **Monitoring level**, all other entries only at level 2 or higher. You set the level under **Settings** on the same page.

If the Software pages show an error, for example **Cannot reach the Scout ELIAS service**, open **System Status** with the colored checkmark icon at the top right. Its **ELIAS backend** entry shows whether UniconOS Management Board reaches the Software Management backend. When the cause is fixed, click **Retry** on the Software page.

Software library

The **Software library** holds the software packages of your devices. The packages are kept in containers, and each container is a folder on the computer where the Software Management

backend runs. Here you create and import containers, add packages to them, and see every file that a container holds.

In the **Navigation** menu, click **Software**, and then click **Software library** in the **Software** menu. Below **Software library**, the menu lists your containers. The search field above the list finds a container by its name. Below the list, **New container** and **Import container** add a container.

When you select a container, its page shows the name of the container, the number of its packages and the date of its last change. Click **Add package** to add packages to it. The page has two tabs:

TAB	WHAT IT SHOWS
Package overview	The packages of the container, grouped by category, and the details of the package that you select
Container explorer	Every file in the container folder, for example packages, signature files and image files

After the installation, the Software library has one container named `elux7`. It is empty until you add packages.

Creating a container

1. Below the container list, click **New container**.

   Software creates a container at once, with a free name such as `New-container-1`. After a few seconds, it selects the new container and opens its name for editing.

2. Type the name of the container, and then click the checkmark or press Enter.

   The name is also the name of the container folder, so it becomes part of the address that devices download from. Use only letters, digits and `. _ + ~ -`, start with a letter, a digit or an underscore, and use at most 127 characters. If a name isn't allowed, Software tells you why.

The container appears in the list under its new name. To keep the name that Software gave it, click the X instead of the checkmark.

> **Note:**

> If the firmware configuration of your devices uses the `__CONTAINER__` parameter in **Path**, give the container the name that the parameter stands for, for example `UC_ELUX7`. See Different eLux versions.

Importing a container

To take over a whole container, for example from another system, import it as a container archive (`.zip`). The archive can hold packages, images, signature files and other container files. Software unpacks it into a new container.

1. Below the container list, click **Import container**.
2. Click **Choose archive**, and select the `.zip` file.
3. In **Name**, check the name of the new container. Software suggests the name of the archive.
4. Click **Import container**.

Software creates the container and uploads the archive. The packages then appear in the list as they are unpacked and checked. After the import, a message lists the files that Software didn't import, and why.

While no container is selected, you can also drag an archive onto the page. The **Import container** dialog then opens with this archive.

You can also import a container that you exported in ELIAS 18. Its packages and its complete images come over, each with its signature files. These parts of an ELIAS 18 container don't come over:

- Templates and incomplete images. ELIAS 18 keeps them in its own formats (`.idt` and `.jidf`).
- The certificates of the container. If you imported certificates in ELIAS 18, import them again, and add your signing certificate, see Certificates.
- The access rights and the change history. Give your administrators the Software rights in UniconOS Management Board, see Software rights.

Adding packages

You add packages as single files (`.epm` and `.fpm`) or as `.zip` archives that hold them, for example the AllPackages archive of a UniconOS version.

1. Select the container, and then click **Add package**.
2. In the **Import package** dialog, click **Choose files**, and select one or more files.
3. Click **Import package**.

The dialog shows the upload progress of each file. Software then imports an archive in the background, and a progress bar on the page shows each step, for example how many package signatures Software has checked. For a large archive, such as the AllPackages archive, the import takes several minutes. The packages appear in the list as they are imported.

If Software refuses a single file, the dialog marks it as **Failed** and shows why. After the import of an archive, a message lists the files that Software didn't import, and why. For example, Software skips these files:

- Files in an archive that aren't packages, signature files (`.sig`) or recovery images (`.dd`)
- Packages that the container already holds under the same file name
- Packages that don't match their signature file
- Files with a name that UniconOS can't use, for example a name with a space
- Archives inside an archive. Upload such an archive on its own.

To add a new build of a package under the same file name, for example a package that you built again, delete the package first, see Deleting a package. For what then happens to the images that use the package, see Images held back after a package change.

If no certificate of the container matches the signer of a package, Software imports the package and shows it as **Signer not trusted**. To change this, import the certificate of the signer, see Certificates.

A container can hold the recovery images (`.dd`) of several UniconOS versions, for example after you import the AllPackages archive of a newer version. Each recovery image takes about 4.4 GB, and recovery sticks use only the newest one. When you no longer need an older recovery image, delete it in the **Container explorer**.

Large uploads need time and disk space:

- While an archive is imported, the drive of the data folder needs free space of about three times the size of the archive.
- An upload must be finished within one hour after it starts. On a slow connection, upload large archives from a computer with a fast connection to UniconOS Management Board.

Package overview

On the **Package overview** tab, the packages are grouped by category, and each category shows how many packages it has. Packages without a category are listed under **Miscellaneous**. Each row shows the name of a package and, below it, its version. Above the list, you find these tools:

- **Package or feature**: finds packages by their name or by the name of one of their features.
- **Filter**: shows only the packages of the categories that you select, or of one **OS edition**. **Remove filter** shows all packages again.
- **Sort packages**: reverses the alphabetical order of the packages in each category.

Click a package to select it. Its row expands and shows the features of the package, see Preselecting features. On the right, the details of the package open:

PART	WHAT IT SHOWS
Version	The versions of the package in the container. **Latest** marks the newest version.
Signature status	Whether a certificate of the container confirms the signature of the package, see Signature status
Package Information	The name, file, copyright, included licenses, build time, package size and **OS edition** of the package
Details	The description, the requirements, and the disk space that the package needs together with its dependencies

The **OS edition** tells you which UniconOS versions a package is built for:

- **<7.2612.0**: UniconOS versions before 7.2612.0
- **>=7.2612.0**: UniconOS 7.2612.0 and later

Packages for versions before 7.2612.0 don't work together with packages for 7.2612.0 and later, so an image can contain packages of only one OS edition. A package can also be built for both OS editions.

SIGNATURE STATUS

The signature status tells you whether Software could confirm the signature of a package with a certificate of the container:

STATUS	MEANING
Signed	A trusted certificate in the container confirms the signature.
Signer not trusted	No trusted certificate in the container matches the signer. Import the certificate of the signer on the **Certificates** page.
Signature invalid	The file doesn't match its signature file (`.sig`), or the signature file is damaged.
Not signed	The container has no signature file for this file.
Not checked yet	Software hasn't checked the signature yet. The status updates by itself.

A warning icon next to a package or a feature means that its file in the container folder was changed after Software approved it, for example because someone replaced the file on the server. Pointing to the icon shows **This package's bytes differ from the approved ones.** Images that use the package are held back, see Images held back after a package change.

PRESELECTING FEATURES

Most packages consist of a main part and features. An image can include a feature or leave it out. When you select a package, its features appear under it in the list. A ticked feature is preselected: when you add this package version to an image, the image gets the feature. In the image editor, you can still change the features of each image.

To change the preselection, tick or clear the features of the package version. UniconOS Management Board saves each change at once.

Some features carry a mark:

- **Required**: an image with this package always gets this feature. You can't clear it.
- **Missing**: the package names this feature, but its file isn't in the container. The details of the package list such features under **Following features are missing**.

Click a feature to see its details, for example its installed size, on the right.

EXPORTING A PACKAGE

To export a package, open the menu of its row, and click **Export**. The download starts at once: a `.zip` archive named after the package and its version, for example `filefetch-1.0.0-1.UC_ELUX7-1.0.zip`, with the `.epm` file of the package, its `.fpm` files and their `.sig` files.

Software exports the version that the row shows. To export another version, select the package, and choose the version under **Version** first. To add the package to another container, for example on another system, import the archive there, see Adding packages.

DELETING A PACKAGE

To delete a package, open the menu of its row, and click **Delete**. If no image uses the package, confirm with **Delete**. If images use it, Software lists them and asks whether you want to delete the package anyway. To delete it, click **Yes**. The package isn't removed from these images. They then show it as missing.

Container explorer

The **Container explorer** tab lists every file in the container folder. Besides packages and features, these are, for example, signature files (`.sig`), image files (`.idf`), the file `container.ini` and recovery images (`.dd`). For each file, the list shows its **File name**, **File extension**, **File size in MB** and **Date**.

- To find files, type a file name or the name of a feature in **File or feature**.
- To open the package or the image that a file belongs to, click its file name.
- To download or delete files, select them. The button **Actions** appears with the number of files that you selected, for example **Actions (2)**. Click it, and then click **Download** or **Delete**. Each file also has its own menu with **Show details**, **Download** and **Delete**.
- Deleting an image file deletes the image. If images use a file that you delete, Software lists them before you confirm.
- A warning icon next to a file name means that UniconOS can't use this name, for example because it contains a space. Software then doesn't use the file. Pointing to the icon shows **Unsafe name** with the reason. Delete the file, rename it in the container folder, or rename its image.

Generating rules

To find the packages that an image needs, Scout uses the dependency rules of the container. Scout rebuilds these rules by itself shortly after the packages of the container change. To rebuild them at once, point to the container in the container list, open its menu, and click **Generate rules**.

Do this, for example, if saving an image reports that the rules of the container are missing or out of date.

Renaming a container

1. Select the container, and then click the pencil icon next to its name.
2. Enter the new name, and then click the checkmark next to it or press Enter.

Software renames the container folder. Devices that download from this container need the new path in their firmware configuration. Don't rename a container while packages are uploaded or imported into it.

Deleting a container

Deleting a container removes its folder with all its packages and images, and it removes the certificates of the container. You can't undo this, so back up the container folder first if you might need its content again.

1. In the container list, point to the container, open its menu, and click **Delete**.
2. In the dialog **Delete container?**, type the exact name of the container in **Name** to confirm.

   The dialog also tells you how many packages and images are deleted with the container.
3. Click **Delete container**.

Software deletes the container and shows **Container deleted**.

If the folder of a container is removed or renamed outside UniconOS Management Board, the container stays in the list with a warning icon. Pointing to the icon shows **Folder missing**, and when you select the container, a banner tells you that its folder is missing. Its certificates, including its signing certificate, are kept. To restore the container, put the folder back under the same name. To remove the container for good, delete it.

Images

An image defines the software of a device: it lists the packages and features that the device installs. Scout keeps each image as an image definition file (`.idf`) in the folder of its container. Once you publish this folder, devices download the image file and its packages from there for a firmware update, see Publishing a container to devices.

In the **Navigation** menu, click **Software**, and then click **Images** in the **Software** menu. The page is titled **Software images**. Below **Images**, the **Software** menu shows the container that you work in

and its images, together with **New image** and **Import image**. The field **Search image** above the images finds an image by its name.

When you open an image, the image editor shows it in two columns:

COLUMN	WHAT IT SHOWS
Defined packages	The packages and features of the image, grouped by category
Package library	All packages of the container. Here you add packages to the image and see their details.

Above the columns, the header shows the file name and the description of the image, its size, the number of its packages and the date of its last change. Marks such as **Signed** show the state of the image, see Marks of an image.

Creating an image

1. In the **Software** menu, below **Images**, select the container.
2. Click **New image**.

   The image editor opens with a new image. UniconOS Management Board gives it a name that is still free.
3. To change the name or to add a description, click the pencil icon next to the name. Enter the name and the description, and then click the checkmark or press Enter.

   The name is also the file name of the image, with the extension `.idf`. It can contain only the letters A to Z and a to z, digits, `.`, `_` and `-`. It must start with a letter, a digit or `_`, and it can have at most 123 characters.
4. In the **Package library**, tick the packages for the image.

   Scout adds the newest version of the package that fits the OS edition of the image, see Package overview. To use another version, open the list of versions of the package and select the version.
5. In **Defined packages**, click a package to show its features. Tick or clear them as needed.
6. Click **Solve dependencies**.

   Scout adds the packages that your packages need, see Solving dependencies. Then the button changes to **Save image**.

7. Click **Save image**.

Scout saves the image and writes its image file, for example `sales.idf`, into the container folder.

If Scout can't write the image file yet, it still saves the image and tells you why. For example, the dependency rules of the container may be out of date. Scout writes the file by itself as soon as it has rebuilt the rules.

Solving dependencies

Packages depend on other packages. When you click **Solve dependencies**, Scout checks the packages of the image against the dependency rules of the container. It adds the packages that are missing and reports the packages that don't fit together, for example packages of different OS editions. For large containers, a solve can take a while. To stop a running solve, click **Cancel** in the progress message in **Defined packages**. If a solve takes too long, Scout stops it and asks you to remove some packages.

Scout adds the newest version of a package that fits, even if it belongs to a newer UniconOS version than the packages that you selected. Check the versions in **Defined packages** before you save the image.

If the solve fails, the message tells you how many conflicts or missing dependencies the image has. Click **Details** to see each problem with a suggestion, for example to remove a package, to keep only one of several packages, or to import a package that the container doesn't hold yet.

Changing an image

To change an image, open it from the image list, and change its packages and features. To remove a package from the image, click the trash icon next to it in **Defined packages**. Then click **Solve dependencies**, and then **Save image**. To discard your changes, click **Cancel**.

If you open another image before you have saved your changes, UniconOS Management Board asks what to do with them. Click **Save and continue** to save them, **Discard changes** to drop them, or **Cancel** to stay in the current image.

While you change an image, Scout locks it for other administrators. Another administrator who changes the same image sees that it is being edited and can't save it. They can still save their version as a copy with **Save as…**. If someone else saved or signed the image after you opened it, UniconOS Management Board doesn't save your changes and offers to reload the image. Reloading shows the current version and discards your unsaved changes.

Further actions for an image

In the image list, each image has a menu. When an image is open, the arrow next to **Save image** opens a menu with some of the same actions. These are the actions:

ACTION	WHAT IT DOES
Export or **Export image**	Downloads the image file (`.idf`), for example to use the image in another system. An image without packages, or with packages that are missing in the container, can't be exported.
Export recovery stick (StickWizz)	Downloads a `.stw` file with the image, its packages and the newest recovery image (`.dd`) of the container. Write the file to a USB stick with StickWizz. This works only for containers of UniconOS 7 that hold a recovery image.
Duplicate	Saves a copy of the image under a new name.
Save as…	Saves the open image, with your changes, under a new name.
Sign	Signs the image, see Signing an image.
Firmware settings	Shows the values to enter in the **Firmware** section of the device configuration, so that devices install this image, see Configuring the devices. Available only for a saved image.
Delete	Deletes the image and its image file. Devices can then no longer download the image.

For how to write a recovery stick, see Writing an image to USB (with StickWizz).

To add an existing image file to the container, click **Import image**, and select the `.idf` file. While no image is open, you can also drag the file onto the page.

An image file that is already in the container folder can have a name that UniconOS can't use, for example a name with a space. Software then doesn't publish, sign or export the image under this

name. Rename the image, and then enter the new file name in **Image file** in the device configuration.

Signing an image

A signature lets your devices check that an image hasn't been changed since you signed it. To sign images, the container needs a signing certificate, see Certificates. Without a signing certificate, **Sign** is greyed out.

1. Make sure that the image is saved.
2. In the image list, open the menu of the image, or click the arrow next to **Save image**.
3. Click **Sign**.
4. If the image contains packages that Scout couldn't verify, UniconOS Management Board asks **Sign with unverified packages?**. To sign the image anyway, click **Sign anyway**.

   The signature of the image then lists these packages, but it doesn't cover their content.

Scout writes the signature file, for example `sales.idf.sig`, next to the image file, and the image is marked **Signed**.

If you change a signed image and save it, Scout removes its signature. Sign the image again after each change. A locked image, an image without packages and an image with packages that are missing in the container can't be signed.

By default, devices don't check signatures. For how to make them check, see Checking signatures.

Images held back after a package change

When you save an image, Software remembers the exact content of its package files. If such a package changes in the container folder afterwards, for example because someone replaced the file on the server, Software holds the image back: it takes the image file (`.idf`) out of the container folder. Devices then keep their installed image and don't install the changed package. Software usually notices such a change within seconds, at the latest within an hour.

While an image is held back:

- In the **Software** menu, a banner above the image list says that images are held back.
- The image has a red warning icon in the image list. Pointing to the icon shows **Held back, a package changed**.
- In the **Software library** and in the **Package library** of the image editor, the changed package has a warning icon.

- Until the change is accepted or the original file is back, you can't sign the image, and you can't export a recovery stick with it.
- If you save the image, UniconOS Management Board stores your changes but doesn't write the image file.

To release the image, do one of the following:

- Put the original package file back into the container folder. Software then releases the image by itself within about a minute.
- Open the image. Under **A package changed on the share**, compare the values under **Approved:** and **Now on the share:**. If the change is intended, click **Accept changes**. Software takes the packages as they are now in the container folder.

Software then writes the image file again. If the image was saved while it was held back, an administrator with the right **Sign images and import certificates** must save it again.

To accept changes, you need the right **Sign images and import certificates** on the container. While an image is held back, only administrators with this right can publish, rename, export or delete it. A package that doesn't match its signature file can't be accepted.

If you add a package under the file name of a package that the container held before, for example a package that you built again, Software treats it as a changed package. The images that use it are held back until someone accepts the change.

Marks of an image

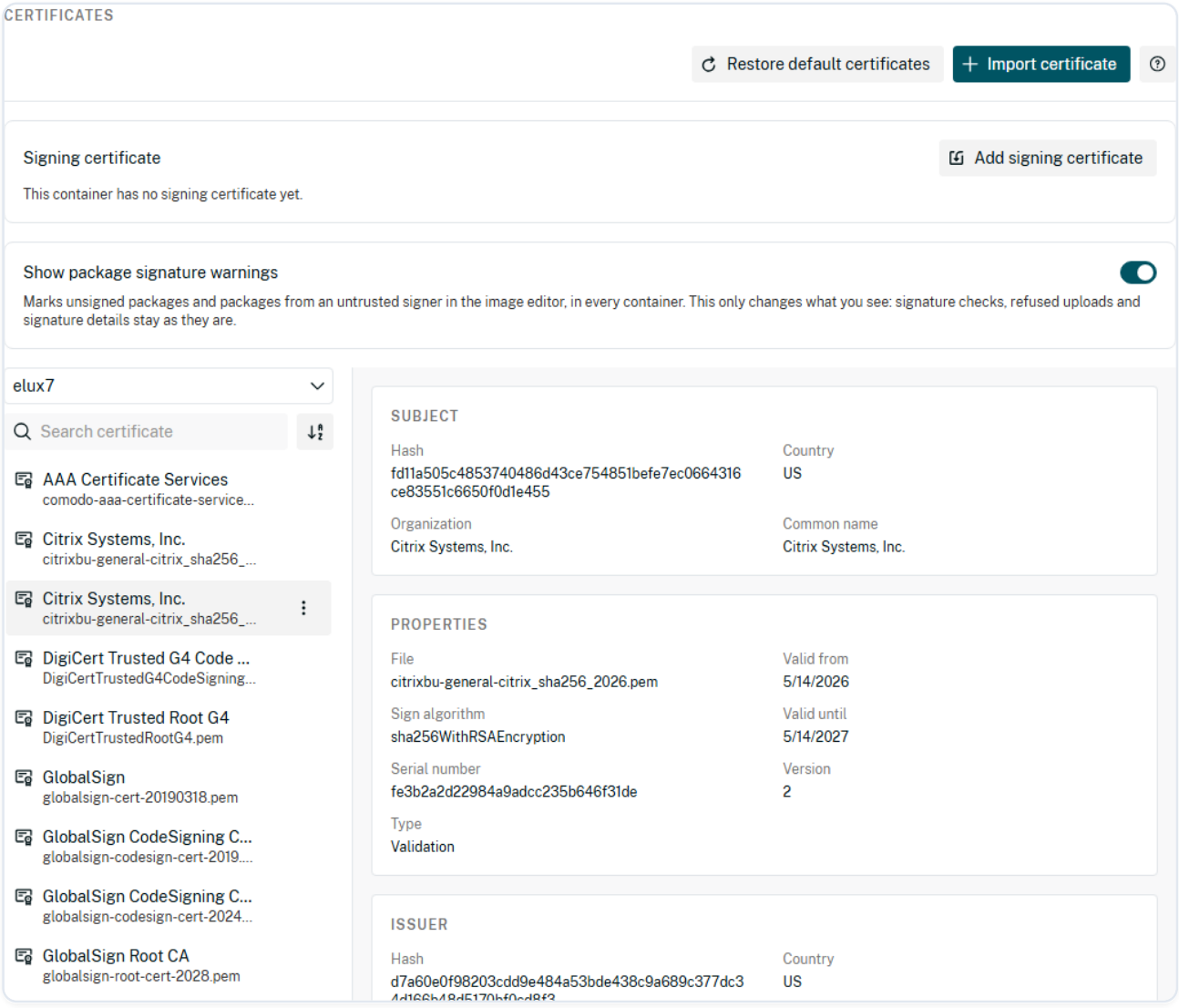
The header of an image shows marks for its state:

MARK	MEANING
Conflicts	Packages of the image don't fit together. Solve the dependencies, and follow the suggestions.
Missing packages	The image uses packages that the container no longer holds. Remove them before you save the image.
Signed	The image has a signature file that matches it.
Locked	The image is read-only. You can export it, but you can't change, copy, sign or delete it.

MARK	MEANING
Template	The image was taken over as a template from an earlier image management, for example ELIAS 18. Software has no functions of its own for templates.

Certificates

Each container has its own certificates. With them, Scout checks the signatures of the packages that you add to the container, and it signs the images that you build in it. In the **Navigation** menu, click **Software**, and then click **Certificates** in the **Software** menu. Then select the container whose certificates you want to manage.



A container has two types of certificates:

TYPE	WHAT IT IS FOR
Validation	Trusted certificates that Scout uses to check the signatures of the packages in the container
Signing	The certificate, with its private key, that signs the images of the container. A container has at most one.

The list shows the signing certificate at the top and marks it with **Signing**. An expired certificate is marked with a dot. Pointing to the dot shows **Expired**. Click a certificate to see its details, for example its subject, its issuer and its validity period.

To import or restore certificates and to add a signing certificate, you need the right **Sign images and import certificates**. To delete a certificate, you need the right **Delete images and packages**, see Software rights.

Default certificates

A new container already has the default certificates. They confirm the signatures of the UniconOS software packages, so these packages show as **Signed** after you add them.

If a default certificate is missing, for example because it was deleted, click **Restore default certificates**. Scout adds the missing default certificates and tells you how many it restored.

Importing certificates

Import a certificate when the default certificates don't cover the signer of your packages, for example for packages that you build yourself. The **Software library** shows such packages as **Signer not trusted**.

1. Click **Import certificate**.
2. In the **Import certificates** dialog, click **Choose files**, and select one or more certificate files in PEM, DER or PKCS #7 format.
3. Click **Import**.

Scout imports the certificates and shows the result for each file. A certificate without an organization name can't be imported.

Adding a signing certificate

To sign images, the container needs a signing certificate with its private key, for example a code signing certificate of your organization. To upload it, open UniconOS Management Board over HTTPS: the browser encrypts the certificate before it sends it, and it does so only over a secure connection.

1. Click **Add signing certificate**.

   If the container already has a signing certificate, the button is **Replace signing certificate**.
2. Click **Choose files**, and select a PFX or P12 file, or a PEM certificate file together with its private key.
3. In **Password**, enter the password of the PFX or P12 file, or of an encrypted private key.
4. Click **Upload**.

Scout stores the signing certificate and shows it at the top of the list. You can now sign the images of this container, see Signing an image.

To check the signature of an image, a device needs the root certificate and your signing certificate in `/setup/cacerts`. By default, devices don't check signatures. For how to make them check, see Checking signatures.

> **Note:**

> Uninstalling the Software Management backend removes the secret that protects the private keys of your signing certificates. After you install the backend again, signing fails, and UniconOS Management Board shows **Could not decrypt the signing key.** To sign again, install the backend with the secret that you noted before you uninstalled it, or add the signing certificate of each container again. For how to keep the secret, see Uninstalling the backend.

Deleting a certificate

1. In the list, point to the certificate, open its menu, and click **Delete**.
2. Confirm with **Delete**.

If you delete the signing certificate, you can't sign images in this container until you add a new one.

Showing package signature warnings

The switch **Show package signature warnings** decides whether the image editor marks unsigned packages and packages from an untrusted signer. It applies to every container, but only to your own view: Scout still checks the signatures and refuses packages that don't match their signature files.

Publishing a container to devices

Your devices don't download their software from UniconOS Management Board. For a firmware update, a device downloads its image file and the packages from a web server or an FTP server. To make the images of a container available to your devices, you publish the container folder on such a server. Then you tell the devices in their device configuration where they find their image.

Scout doesn't set up this server for you. For HTTP and HTTPS, use a web server that delivers static files, for example IIS or nginx. For FTP and FTPS, use an FTP server. For the steps with IIS, see Example of publishing the containers with IIS.

The container folders

Each container is a folder with the name of the container. By default, the container folders are in `C:\ProgramData\Unicon\Software\containers` on the computer where the Software Management backend runs. If another data folder was chosen during the installation, they are in its `containers` subfolder.

All files that devices need are side by side at the top level of the container folder:

FILE	CONTENT
`.epm`, `.fpm`	The packages and their features
`.sig`	The signature files of the packages, and of the signed images
`.idf`	The image files
`container.ini`	Information that identifies the container

When you save an image, Scout writes its `.idf` file into the container folder. When you sign the image, Scout also writes its `.idf.sig` file. When you delete the image, Scout removes both files.

If a package changes in a container folder after you saved an image with it, Software takes the image file out of the folder until the change is accepted or the original file is back. Devices then find no image file and keep their installed image, see Images held back after a package change.

Make sure that no unauthorized person can change the files in the data folder.

Using your existing container folder

If you used the legacy ELIAS, a web server may already publish your containers to the devices. You can let Software use this folder instead of its own `containers` folder. Software then lists your existing containers, packages and images, and your devices keep their firmware settings. Software doesn't change any file in the folder until you edit an image or a container in UniconOS Management Board.

To use your existing folder:

1. On the computer where the Software Management backend runs, open `config\.env` in the program folder of the backend. By default, this is `C:\Program Files\Unicon\Scout\SoftwareManagement\config\.env`. Use an editor that runs as administrator.
2. Add these two lines. In `CONTAINERS_DIR`, enter the folder that holds your container folders, for example `C:\inetpub\wwwroot\eluxng` or `\\fileserver\eluxng`:

   ```
   CONTAINERS_DIR=C:\inetpub\wwwroot\eluxng

   FORCE_RESCAN=true
   ```

3. Restart the service **Software Management backend**.
4. Wait until Software has read the folder. Then open **Software** > **Software library**. Your containers are listed with their packages and images.
5. Remove the line `FORCE_RESCAN=true` again. With this line, Software reads all containers again at every start, which takes a long time.

Keep these points in mind:

- The Software Management backend must be able to read and change the files in the folder. If the folder is on another computer, give the computer account of the backend computer read and write access to it.

- Older containers keep their packages in a `Packages` subfolder. Software doesn't read the files in this subfolder. Create such a container again in Software, for example by importing the AllPackages archive.
- Containers whose folders aren't in this folder, for example the container `elux7` that the installation created, stay in the list and are marked **Folder missing**. Delete them if you don't need them.
- Don't use the root of the system drive, `C:\`, as the folder. Software would list the Windows folders as containers.
- A repair or an update of Scout keeps the line `CONTAINERS_DIR`.
- The rules in Setting up the server still apply.

Setting up the server

Set up the server so that devices can read the container folders, and so that nobody can change them through it:

- Publish the `containers` folder, or the folder of a single container, read-only.
- Make sure that the server delivers files with the extensions `.epm`, `.fpm`, `.idf` and `.sig`.
- For HTTPS, use a server certificate that your devices trust.
- For FTP or FTPS, point the FTP server to the same folder.

The devices must always get the current files of the container folders. A copy or a cache could still offer an image file that Software took out of a container folder, so these rules apply:

- Let the devices download directly from the container folders, through a server that reads the files for every request. On IIS, this needs a change in the caching settings, see Example of publishing the containers with IIS.
- Don't use copies of the container folders, for example mirrors or replicas that DFS Replication, rsync or scheduled robocopy jobs keep up to date. Such copies aren't supported. Software doesn't watch them, so it doesn't notice a package that was changed there.
- If a second web server or a caching proxy is between the devices and the container folders, it must read each file from the container folder for every request. It must not keep copies of image files (`.idf`).

Devices can't download from the Software Management backend itself. The backend serves only UniconOS Management Board.

Example of publishing the containers with IIS

If a web server already publishes your containers from the legacy ELIAS, you can keep it, see Using your existing container folder.

Internet Information Services (IIS) is the web server of Windows Server. If you don't have a web server for your devices yet, install IIS on the computer where the Software Management backend runs. By default, this is the Scout Board computer. The following steps publish the `containers` folder under the address `http://<server>/eluxng`, where `<server>` is the name of this computer:

1. Find the `containers` folder. By default, it's `C:\ProgramData\Unicon\Software\containers`. If another data folder was chosen during the installation, it's the `containers` subfolder of that folder. Each container is a folder in it, with the name of the container.
2. Install IIS. In PowerShell as administrator, run this command:

   ```
   Install-WindowsFeature Web-Server -IncludeManagementTools
   ```

3. Open the **Internet Information Services (IIS) Manager**. Under **Sites**, right-click **Default Web Site**, and click **Add Virtual Directory**.
4. In **Alias**, enter a name for the address, for example `eluxng`. In **Physical path**, select the `containers` folder. Click **OK**.
5. Let IIS deliver the files of UniconOS. Select the new virtual directory `eluxng`, and double-click **MIME Types**. For each of the extensions `.epm`, `.fpm`, `.idf` and `.sig`, click **Add**, enter the extension in **File name extension** and `application/octet-stream` in **MIME type**, and click **OK**.

   Don't add `.*` for all extensions instead. IIS would then deliver every file in the folder.
6. Give IIS read access to the `containers` folder. In a command prompt as administrator, run this command. If your `containers` folder is in another place, use its path.

   ```
   icacls "C:\ProgramData\Unicon\Software\containers" /grant "IUSR:(OI)(CI)RX" "IIS_IUSRS:(OI)(CI)RX"
   ```

7. Turn off caching for image files and signature files. Select the virtual directory `eluxng`, and double-click **Output Caching**. Click **Add**, and enter `.idf` in **File name extension**. Select

User-mode caching and **Kernel-mode caching**, select **Prevent all caching** for both, and click **OK**. Add the same rule for `.sig`.

This step is required. With a cache, IIS could still deliver an image file that Software has taken out of the container folder.

8. Test the address. On another computer, open `http://<server>/eluxng/<container name>/<image name>.idf` in a browser, for example `http://scout01.example.com/eluxng/elux7/sales.idf`.

   The browser downloads the image file. If it can't connect at all, check that the firewall of the IIS computer allows HTTP. If it shows the error 401 or 403, IIS has no read access to the `containers` folder. If it shows the error 404, check the address. To see the exact error, open the address in a browser on the IIS computer: the error 404.3 means that the MIME types are missing.

9. In the device configuration, open the **Firmware** section, and enter these values:

   - **Protocol**: **HTTP**
   - **Server**: the name or IP address of the IIS computer. Don't enter the port of the Software Management backend (`22170`). Add a port, for example `scout01.example.com:8080`, only if IIS doesn't use port 80.
   - **Path**: `eluxng/<container name>`, for example `eluxng/elux7`, without a slash at the start
   - **Image file**: `<image name>.idf`, for example `sales.idf`
   - **User** and **Password**: leave them empty.

   The line **Entered path** below the fields shows the whole address. It must be the same address that worked in the browser.

10. Restart the devices, so that they get the new configuration. Then, to install the image, send the command **Update software** to the devices.

Keep these points in mind:

- The container must match the UniconOS version of your devices.
- If the container folder has no `.idf` file for your image, open the image in **Software**, and save it. If the image uses packages that are missing in the container, remove them first.
- IIS delivers the files straight from the container folders. After you change and save an image, you don't need to copy anything.

- IIS delivers only files whose extension has a MIME type. If your devices download other files from the container, for example the files of a migration from eLux® RP 6, add MIME types for these extensions too.
- If your devices check signatures, sign the image first, see Checking signatures.

Configuring the devices

In the device configuration, the **Firmware** section tells the devices where to find their image. Enter the address of your server and of the image there:

SETTING	TYPE	WHAT TO ENTER
Protocol	Dropdown	The protocol of your server: **HTTP**, **HTTPS**, **FTP** or **FTPS**
Server	Text field	The name (FQDN) or IP address of the server
Path	Dropdown	The path of the container folder on the server. If the server publishes the `containers` folder as its root, this is the name of the container, for example `elux7`.
Image file	Dropdown	The file name of the image, for example `sales.idf`

Software shows these values for each saved image. In the image list, open the menu of the image, and click **Firmware settings**. Click the copy button next to a value to copy it, and paste it into the device configuration.

Firmware settings shows **Protocol**, **Server** and the whole address of the image file (**Full address**) only if Software knows the address under which your server publishes the `containers` folder. Until you set this address, the dialog shows only **Path** and **Image file**, and a hint tells you where to set the address. To set it:

1. On the computer where the Software Management backend runs, open `config\.env` in the program folder of the backend. Use an editor that runs as administrator.

   By default, the program folder is `C:\Program Files\Unicon\Scout\SoftwareManagement`.

2. Add the line `DEVICE_DOWNLOAD_URL=` with the address of the `containers` folder on your server, for example `DEVICE_DOWNLOAD_URL=http://scout01.example.com/eluxng` for the IIS example above.

   The address starts with `http://`, `https://`, `ftp://` or `ftps://`. It can't contain a user name or a password.

3. Restart the service **Software Management backend**.

If the address isn't valid, the backend doesn't start, so check the address carefully. An update of Scout keeps the line.

To offer the names of your images in the **Image file** list, add them under **Preferences** > **Predefined firmware configuration**, see Preferences.

For the other settings of the section, see Device configuration / Firmware.

A device gets its changed configuration when it restarts. It installs the image with its next firmware update, for example when you run the **Update software** command, or when it checks for a new image version on start.

Checking signatures

Under **Security**, in the **Firmware** section of the device configuration, you decide which signatures the devices check before an update: **Image definition files (.idf)** and **UniconOS software packages**.

By default, both check boxes are cleared. The devices then check no signatures. They install an image and its packages even if these are unsigned or were changed on the server. To make the devices refuse unsigned or changed images and packages:

1. Sign your images, see Signing an image.

   An image has a signature file only after you sign it. The signature files of the packages come with the packages.

2. Make sure that the devices have the root certificate and your signing certificate in `/setup/cacerts`, see Firmware security through signatures.

3. In the **Firmware** section of the device configuration, under **Security**, select **Image definition files (.idf)** and **UniconOS software packages**.

If a device checks the signatures of image files, it doesn't install an unsigned image: the update fails, and the device keeps its installed image.

A device checks the signatures of packages only if it has an update partition. A device without an update partition checks only the signature of the image file. For details, see Device configuration / Firmware.

Preferences

On the Navigation Panel, on the bottom section,you will find the **Preferences** entry with preferences for various functions.

Devices

General device and filter settings, especially with regard to onboarding devices

1	OU new devices are assigned to, by default
2	Activates the OU filter for new devices. The OU filter has priority over other methods. The OU filter is configured using subnets and masks or using specific asset information of the devices.
3	Newly added devices are not synchronized with the server's device configuration
4	Allows dynamic assignment of devices via DHCP
5	The Scout Server accepts only devices with registered MAC addresses.

The setting **Devices that join a protected or hidden OU without sending its password to Scout Server** decides what Scout Server does when a device asks to join an OU that has an access password, or a hidden OU, without the password:

- **Reject** (default, also after an upgrade): Scout Server places a new device in the default OU. A device that moves itself to another OU stays in its current OU.
- **Accept and log**: The device joins the OU, as in earlier versions, and Scout Server logs each such join.

Scout Server refuses a wrong password with both settings. It checks the password only when a new device joins an OU or when a device moves itself to another OU. Devices that are already in an OU, moves by an administrator and OU filter rules aren't affected. A change of the setting applies without a restart of Scout Server.

The setup wizard of UniconOS 2611 and earlier checks the password on the device, but doesn't send it to Scout Server. Devices that get their OU from DHCP options, from a GroupID in their image or from Scout Cloud Gateway don't send it either. With **Reject**, such new devices start in the default OU. Place them in the protected OU with an OU filter rule, or move them there after they have joined. To keep the behavior of earlier versions, select **Accept and log**.

Logging

On the Scout Server machine, server activities of the Scout Server and the Scout Keep Alive service are logged.

1	Log level **Error**: Only error messages, **Info**: Error messages and additional information (default), **Debug**: Log files with extended content
2	Configure log file rotation / File size
3	Configure log file rotation / Number of files
4	Enable Scout Server logging (`eluxd.log`)
5	Enable Keepalive logging (`keepAlive.log`)

Predefined commands

User-defined commands can be centrally predefined and provided as ready-to-use commands for administrators managing their devices remotely. All active commands in this list can be applied by administrators to individual devices, to OUs, or to Dynamic Device Groups.

A predefined command is shared with all administrators by default, but can be restricted to specific groups. For the executing administrators, the name of the command is displayed instead of the command.

1	Add new entry
2	Delete selected entries
3	Name of a command
4	Command definition (syntax)
5	The command requires system rights
6	The command will be displayed in the list-field for predefined commands.
7	Restrict command to specific administrators or administrator groups
8	Edit the name or syntax of an entry

To create a predefined command, click **Add** (1), and then in the new entry, specify the **Name** (3), **Command** (4) and further options (5, 6, 7).

Firmware preferences

For **Device Configuration** > **Firmware**, you may predefine values that operational administrators are allowed to use. If you restrict the administrators' object rights so that only predefined entries can be selected, no custom values can be created. For further information, see Protecting firmware configuration in the **Scout** guide.

The following values can be predefined:

IMAGE TYPES	DESCRIPTION AND EXAMPLE
Image files	Names of images which may be used for firmware updates and upgrades
	Example: `recovery_x.idf`
Container for software packages and images	Container paths which may be used to select images
	Example: `elias/UC_RC6_X64`

IMAGE TYPES	DESCRIPTION AND EXAMPLE
UEFI files	Names of `.udf` files which may be used to update the devices UEFI firmware with matching binary data.
	Example: `uefi.udf`

To create a predefined firmware value, click **Add**, and then in the new entry, enter the exact name or path. The entry must correspond in spelling to the existing file and path names. Note the following:

- File and path names are case-sensitive
- Do not use spaces.
- Specify file names with their extensions such as `.idf`.
- An image file name can contain only letters, digits and `. _ + ~ -`. It must start with a letter, a digit or `_`, and it can have at most 127 characters.

The **Active** option allows you to specify for each defined entry whether it will be displayed or hidden in the **Firmware** dialog.

To change a list, you need its right, for example **Administration of predefined IDF** for the image files, see Existing rights UniconOS Management Board now enforces.

OU filter

If **Assign OU depending on filter settings** is on, **Filter Type** chooses how Scout assigns new and reconnecting devices to an OU: **Subnet filter** (by device network address) or **User-defined filter** (by configured asset information).

With **User-defined filter**, the table's toolbar offers two ways to create a filter:

- **Add filter** creates a typed filter: an expression such as `ELUX_HOSTNAME=pc-01`, matching the syntax used in the legacy Scout Console. For the full syntax and the list of `ELUX_` variables, see OU filter as a user-defined filter.
- **Add rule-based filter** opens a visual rule builder. Each condition has **Property**, **Operator** and **Value**; link conditions in a block with **AND** or **OR**, and add further blocks with the **AND / OR**

control between them. The dialog shows the generated expression and validates the rule before adding it. A **Destination OU** is required for each rule-based filter.

  - **Property** offers the device properties also used by typed filters (such as Hostname, IP Address, Device Type), plus **In Subnet (IP Check)** (value: a CIDR network, for example `192.168.1.0/24`) and **Custom Property** (a property name your own custom package adds).
  - Rule-based filters can only match `ELUX_` properties and customer-defined `_CUSTOM.` properties from a custom eLux package. Server-side `SCOUT_` properties are not available at OU-assignment time.
  - Operators for ordinary properties are **Contains**, **Equals**, **Starts With**, **Ends With** and **Regex**; matching ignores case. Round brackets cannot be used in a value with any operator — for alternatives, write `t640|t740` instead of `(t640|t740)`, or add one condition per alternative linked with OR.

- Typed and rule-based filters are listed together and evaluated in one **Sequence** order, lowest first; the first active filter that matches decides the OU. If a typed and a rule-based filter share the same Sequence and both match, the typed filter wins.
- Both the **Subnet filter** and **User-defined filter** tables have a **Search filters** field (a plain contains-search, not case-sensitive) and row-per-page options of 25, 50, 75, 100 or All, remembered per table.

Device names

A device gets its name from a name template, applied the first time it contacts Scout. The template can use the device's hardware properties, so names can reflect the serial number, manufacturer, model or asset tag instead of only the MAC address, IP address or a counter.

1	Turn on **Use name template**, then enter a template in **Name template**
2	**Insert variable** adds the selected variable, without arguments, at the cursor position
3	**Sample** shows the name the template gives an example device

Click a variable in the **Insert variable** menu to add it to the template:

VARIABLE	SHOWS
`<MAC>`	The MAC address (12 characters)
`<IP>`	The IP address
`<COUNT>`	The lowest number that no other device name uses
`<SERIAL>`	The serial number
`<SUPPLIER>`	The manufacturer
`<DEVICETYPE>`	The model
`<ASSETTAG>`	The chassis asset tag, or the baseboard asset tag if the chassis has none

Each variable takes up to four optional, comma-separated arguments: `<VARIABLE,from,to,separator,case>`.

- **from** / **to**: the first and last character to use, counting from 1. A value past either end is moved to the nearest valid position, so `<SERIAL,1,255>` is the whole serial number.
- **separator**: inserted after every second character. For `<IP>`, this is the character between the parts of the address.
- **case**: `U` for upper case, `L` for lower case.
- `<COUNT,start,max,fill>` uses a different set of arguments: a number from *start* (default 0) up to *max* (default 99999), padded with the *fill* character (default 0). Scout takes the lowest number in this range that no other device name uses, also after Scout Server restarts. A number that becomes free, for example when you delete a device, is used again after Scout Server restarts or after you save the device name settings.

An argument left empty counts as zero, not as its default. For example, `<SERIAL,,,,U>` gives only the first character; for the whole serial number in upper case, write `<SERIAL,1,255,,U>`.

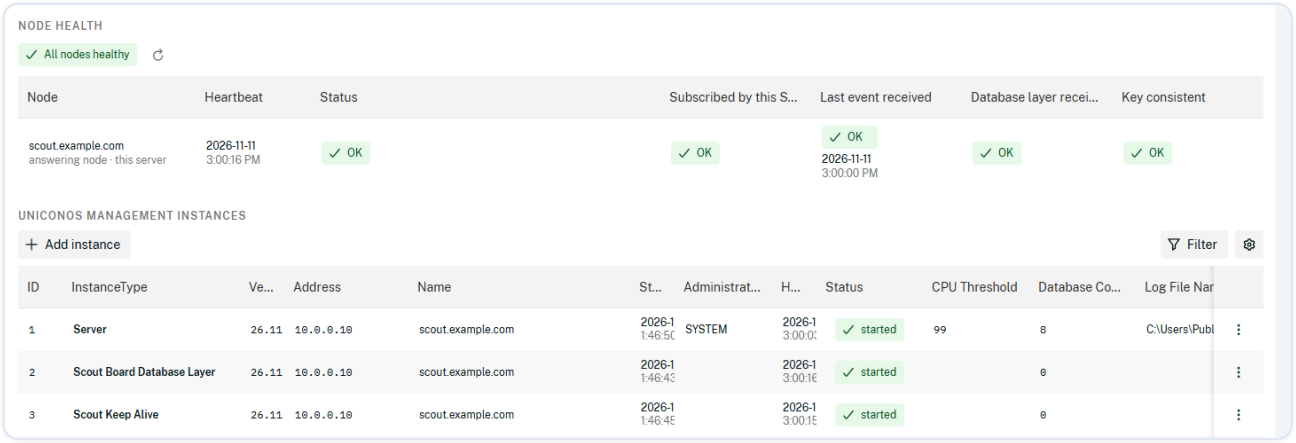
	Template `PC-<SERIAL,1,6,,U>` gives the sample device the name **PC-5CG123**

A template can be up to 255 characters long. After the variables are filled in, every character other than a-z, A-Z, 0-9 and the hyphen becomes a hyphen, and a hyphen at the start or end of the name becomes the letter A. A variable whose value the device does not report expands to nothing.

- **Apply name template only to new devices** (on by default): a new device is named from the template. With this off, a device is also renamed at its next check-in if its name no longer matches the template, except for templates containing `<COUNT>`.
- If the template would give a device a name that another device already has, Scout does not apply it to that device. A registered device keeps its current name.
- A template can also be set for a single OU, see **Advanced device configuration** > **Management** in that OU's context menu. Turn on **Use parent** to use the template for all devices, or enter a template for that OU only.

UniconOS Management instances

UniconOS Management instances lists the Scout components that are registered in your Scout database, and it shows whether your Scout Servers work together. You need this page mainly when several Scout Servers share one Scout database. In the **Navigation** menu, click **Preferences**, and then click **UniconOS Management instances**. To see this entry, you need the menu right **Menu : View -> Scout entities...**. UniconOS Management Cloud doesn't offer it.



The page has two parts: **Node health** at the top and the list of instances below it.

Node health

Node health shows each Scout Server that uses the Scout database, and it checks whether the UniconOS Management Board that you're signed in to receives the events of this server. UniconOS Management Board needs these events, for example, to show status changes of devices without a reload and to set up mirroring sessions. With a single Scout Server, the table has one row.

Above the table, a summary shows the overall result: **All nodes healthy**, **Some nodes have warnings**, **At least one node is failing** or **Not all information is available**. To check again, click the refresh icon next to it.

Each row of the table is a server. The columns show these values and checks:

COLUMN	SHOWS
Node	The host name of the server. Below it, **this server** marks the server that this UniconOS Management Board runs on, and **answering node** marks the server whose Database Layer this UniconOS Management Board uses.
Heartbeat	When the Database Layer of the server last reported to the Scout database.
Status	The overall result for the server. If a check fails or warns, its hint appears below the result.
Subscribed by this Scout Board	Whether this UniconOS Management Board receives the events of the server on port 22151.
Last event received	When this UniconOS Management Board last received an event from the server. The Database Layer sends an event at least every 30 minutes, so the check warns after 61 minutes without one. If no event has arrived yet, the check shows **Unknown** during the first hour after UniconOS Management Board has connected to the server.
Database layer receiving from Scout Server	Whether the Database Layer of the server receives the events of its Scout Server. The check fails, for example, if the Database Layer can't reach its Scout Server, if Scout Server has stopped, or if the Database Layer hasn't sent a heartbeat for more than 2 minutes. A Scout Server without device activity sends no events, so a long silence alone doesn't fail the check.

COLUMN	SHOWS
Key consistent	Whether the Database Layer of the server uses the current Scout Server key from the Scout database.

The status and each check show **OK**, **Warning**, **Failing** or **Unknown**. To see the hint of a check, point to it, or move the keyboard focus to it. The hint tells you what to do, for example which server refuses this UniconOS Management Board and which setting to change.

Each UniconOS Management Board shows only its own connections. Therefore, check **Node health** in the UniconOS Management Board of every server.

If the Database Layer of a server doesn't run while its Scout Server runs, UniconOS Management Board keeps the server in the table, with a warning that no running database layer is registered for it. To clear the warning, start the **Scout Board Database Layer** service on this server.

For causes and solutions, see Troubleshooting in **Scout Server cluster**.

List of instances

Below **Node health**, the list **UniconOS Management instances** shows every Scout component that is registered in the Scout database, for example each Scout Server, UniconOS Management Board, Scout Board Database Layer and Scout Keep Alive service. For each instance, the list shows the type, the version, the address and name, the start time, the heartbeat and the status, **started** or **stopped**.

To add a load balancer, click **Add instance**, see ManagerLoadBalancing by dedicated load balancer. With the menu at the end of a row, you can edit or delete an instance. You can delete only a stopped instance. The entries of UniconOS Management Board, the Scout Board Database Layer and the Scout Keep Alive service can't be edited or deleted.

Enrollment settings

Enrollment settings is where you create enrollment tokens and manage device certificates for devices that connect to UniconOS Management Cloud. On-premises installations don't show this entry. For details, see Enrollment settings.

Administrator rights in UniconOS Management Board

UniconOS Management Board uses the same administrator records as the legacy Scout Console, but it adds its own base rights and object rights for functions that only exist in UniconOS

Management Board. Go to **Security & permissions** > **Manage administrators**, select an administrator, then **Rights management**.

In **Base rights**, **Menu rights** and **Base object rights**, you can change only rights that you hold yourself — this already applied when editing an existing administrator, and **Duplicate admin** gives the new administrator only the rights you hold. An administrator created in the Scout Console with minimum access, and never edited since, is denied the **Rules** rights and the four rights for running actions described below.

Dynamic Configuration rights

Dynamic Configuration in **Base rights** gates any access to **Configuration labels**, **Configuration templates** and **Configuration rules**. Expand it for the individual rights:

RIGHT	CONTROLS
View labels	Seeing the labels list and the configuration templates, and exporting templates
Create labels / Edit labels / Delete labels	Adding and duplicating, editing, deleting labels and configuration templates. Importing a template needs **Create labels**.
View rules	Seeing the rules list
Create rules / Edit rules / Delete rules	Adding, editing, deleting rules
Activate / deactivate rules	The **Active** switch in the rules list; saving the change also needs **Edit rules**
Reorder rules	Dragging rules and **Send to Top** / **Send to Bottom**; saving the new order also needs **Edit rules**

To give an administrator read-only access to labels or rules, allow **Dynamic Configuration** and **View labels** (or **View rules**), and deny the rest. Denying **Dynamic Configuration** hides all three pages.

Rights for running actions

Four rights control how an administrator runs Scout commands:

RIGHT	WHERE	CONTROLS
Execute action on multiple devices	Base rights	**Execute action** on an OU, a Dynamic Device Group, or more than one selected device — even when only one device is selected in that view
Execute action immediately	Base object rights > Schedule action	The schedule type **Now**
Schedule action (once)	Base object rights > Schedule action	The types **Once** and **On Next System Start**
Schedule action (recurring)	Base object rights > Schedule action	The type **Repeat every**

Running an action also needs the command right for the target action itself (for example **Restart**, under **Execute action**). Only **Base object rights** count for the three schedule rights — customizing an OU's or a device's own object rights has no effect on them.

Without **Execute action on multiple devices**, **Execute action** is greyed out for OUs, Dynamic Device Groups and multi-device selections, with the tooltip **You cannot execute actions on more than one device at once**. The administrator can still act on a single device from that device's own menu.

Software rights

The **Software** section of the **Navigation** menu has its own group in **Base rights**. **Software** gates the whole section: without **Software** or **View software**, an administrator doesn't see **Software** in the **Navigation** menu. Expand **Software** for the individual rights:

RIGHT	CONTROLS
View software	Seeing containers, packages, images and certificates, exporting images and packages, showing the firmware settings of images, and downloading files from a container
Upload and edit content	Adding packages, and creating, importing, duplicating, changing and saving images

RIGHT	CONTROLS
Delete images and packages	Deleting packages, files, images and certificates. Deleting a container also needs **Manage containers**, and deleting the signing certificate of a container also needs **Sign images and import certificates**.
Manage containers	Creating, renaming and deleting containers, adding, replacing or deleting the file `container.ini` of a container, and changing the preselected features of packages. Importing a container also needs **Upload and edit content**.
Run solver	**Solve dependencies** in the image editor and **Generate rules** for a container
Sign images and import certificates	Signing images, importing certificates, restoring the default certificates, adding or replacing the signing certificate of a container, and accepting changed packages. While an image is held back after a package change, only administrators with this right can publish, rename, export or delete it, see Images held back after a package change.

An action that the administrator's rights don't allow stays visible on the Software pages, but it is greyed out. Pointing to it shows **You do not have the rights for this. Ask an administrator.**

Changes that can give an administrator more Software rights need a sign-in from the last 5 minutes:

- Allowing a Software right under **Rights management**
- Adding or duplicating an administrator who gets Software rights
- Choosing a container role with more rights than before, or **None**
- Changing **Rights consolidation**
- If the rights of AD groups are consolidated: deleting the object rights of an OU or a device, and changing the root OU of an administrator

If you signed in earlier, UniconOS Management Board doesn't save the change and shows **This change can give administrators more Software rights, so it needs a sign-in from the last 5 minutes. Sign in again, then repeat the change.** Click **Sign in again**, sign in, and repeat the change. Changes that only take rights away, and deleting an administrator, don't need a new sign-in. With Microsoft Entra ID and OpenID Connect, this works only if the app registration of Scout Board includes the optional claim `auth_time`, see Configuring authentication options for UniconOS Management Board.

For what the Software section does, see Software.

Roles per container

With the rights above, an administrator has the same access to every container. To give an administrator less access to a single container, assign a container role. A role can only narrow the administrator's Software rights, never extend them.

1. Go to **Security & permissions** > **Manage administrators**, select the administrator, and then click **ELIAS containers**.

   This section appears only while the **Software** section is available to you.
2. Click **Edit**.
3. For each container, select a role.

   UniconOS Management Board saves the role at once and shows **Container role saved**. The administrator gets the changed rights within about a minute, also on pages that are open already.

These roles are available for a container:

ROLE	RIGHTS ON THE CONTAINER
None	No role: the administrator's Software rights from **Rights management** apply.
Info	**View software** only. The administrator can see the container but can't change anything in it.
User	**View software**, **Upload and edit content** and **Run solver**. The administrator can add packages and build images, but can't delete anything, manage the container or sign.

ROLE	RIGHTS ON THE CONTAINER
Admin	All Software rights

These rules apply to container roles:

- A role takes effect only for administrators who hold **Software** and **View software** under **Rights management**. For other administrators, you can't select a role.
- In the role list, a role that goes beyond the administrator's Software rights is marked **(exceeds the ELIAS permissions)**.
- You can assign roles only for containers on which you hold **Manage containers** yourself, and you can't hand out a right that you don't hold on that container. Such roles are greyed out and marked **(beyond your own permissions on this container)**.
- You can't remove a role with **None** if the administrator would then get rights on the container that you don't hold there yourself. **None** is then greyed out and marked **(clearing would hand back permissions you lack on this container)**.
- A container entry that matches none of the roles is shown as **Custom**. You can replace it with a role, or remove it with **None**.
- If an administrator signs in through AD groups, the container roles of the groups are combined in the same way as their other rights, as set in **Rights consolidation**.
- Creating and importing a container depend only on **Rights management**, because the container doesn't exist yet.
- If a container role narrows your own Software rights on a container, you can't give another administrator Software rights that let them do more on this container than you can, unless their own container role keeps them out. This applies when you add, duplicate or change an administrator. The rights editor offers only the Software rights that you may grant, and it shows the others as locked. If UniconOS Management Board can't read the rights that it needs for this check, it refuses the change.

Existing rights UniconOS Management Board now enforces

UniconOS Management Board also honours several rights that previously had no effect in UniconOS Management Board:

FUNCTION	RIGHT NEEDED
Download log files (system menu)	Menu rights **Menu : View -> System diagnostic** and **Menu : View -> System diagnostics -> Server log**
Command History	Menu right **Menu : View -> Update history**. Purging the command history also needs the base right **Execute action on multiple devices**.
Go to origin, for an inherited base application	Base right **Edit base application**
Export / Import in the OU menu	Object rights **Export** + **Export of the OU tree**; **Import** + **Import of an OU tree**
Export devices / Import devices	Menu rights **Menu : File -> Export -> Devices**; **Menu : File -> Import -> Devices**
Reactivate OU filter	Menu right **Devices** under **Menu : Options -> Advanced options**
Import a base configuration or an advanced configuration	Rights on the import's target, plus the right for every section of the imported file that applies to this target. Sections that don't apply, for example keyboard settings in an import into the base configuration, are skipped and need no right. If any section is denied, nothing is imported, and the dialog shows **You do not have permission to import configurations.**
Changing predefined commands	**Administration of predefined commands** — the same right that shows predefined commands in UniconOS Management Board
Changing the predefined image files, update paths and UEFI files under **Preferences** > **Predefined firmware configuration**	**Administration of predefined IDF**, **Administration of predefined update paths** or **Administration of predefined UEFI files**, for the list that you change
Recovery settings under **Preferences**	Menu right **Menu : Options -> Recovery settings**

FUNCTION	RIGHT NEEDED
UniconOS Management instances under **Preferences**	Menu right **Menu : View -> Scout entities...**
Changing the device settings under **Preferences** > **Devices**, for example the device names and the options for new devices, and reading or changing them through the REST API (`/api/v1/settings/devices`)	Menu right **Devices** under **Menu : Options -> Advanced options**
The **Activity log** and **Rights consolidation** under **Security & permissions** > **Manage administrators**	**Edit administrators**
Choosing the diagnostic templates when you request the diagnostic files of a device	Object right **Select diagnostic templates** on the device. Without it, the device sends the default diagnostic files.
Using the UniconOS Command Scheduler through the REST API	**Device scheduler**
Setting INI file entries through the REST API	**Advanced file entries**
Exporting the OU structure through the REST API	Object rights **Export** + **Export of the OU tree**
Importing an OU structure through the REST API	Object rights **Import** + **Import of an OU tree**

If an administrator's existing rights already deny one of these functions, UniconOS Management Board now hides or blocks it, where it previously allowed it. Through the REST API, a request that the administrator's rights don't allow gets HTTP 405, see Using the REST API.

An administrator can delete only the Dynamic Device Groups they created themselves, not ones created by another administrator.

A user-defined command saved in the **UniconOS Command Scheduler** needs the device password, wherever it is saved from. In UniconOS Management Board, the **Import** dialog for a configuration or an OU asks for it when the imported file contains such a command: **This file contains a user-defined command. Confirm the device password to import it.** Through the

REST API, the write needs a `devicePassword` field; a missing or wrong password gets HTTP 403.

Rights on the devices and configurations that an action reaches

UniconOS Management Board checks the rights on each device and each configuration that an action actually reaches:

- A command, and a request for support files or diagnostic files, needs your rights on the device that it reaches. If you send a command to several selected devices and you don't have the right for it on one of them, UniconOS Management Board refuses the command for all of them.
- A command to a Dynamic Device Group needs your right for this command on every device in the group. If the group contains a device that you can't see, or on which you don't have this right, UniconOS Management Board refuses the command for the whole group. You can send commands only to your own Dynamic Device Groups.
- You can read and save the configuration of an OU or a device only if you have the needed rights on this OU or device. The save changes only this configuration, never the base configuration or the configuration of another OU or device. This also applies to the sections of the advanced device configuration in the REST API.
- With the right **Advanced file entries** on an OU or a device, you can change only the INI file entries of this OU or device.

If you change the rights of an administrator who is signed in, the change applies to their next action. They don't need to sign in again.

Granting rights

From UniconOS Management 2611, the rule at the top of this page applies wherever you change rights: you can't give an administrator a right that you don't hold yourself. In detail:

- At an OU or a device, you can grant only the rights that you hold at this OU or device, and at every OU and device below it where your own rights are restricted.
- **Reset object rights** is refused if the OU or the device would then inherit a right that you don't hold there.
- If an OU or a device inherits rights that you can't grant, UniconOS Management Board refuses to create object rights for it, instead of silently removing these rights.
- Only administrators with **Change menu rights** can change menu rights.

- If your rights are limited to a root OU, you can set the root OU of an administrator, your own included, only to this OU or an OU below it. You also can't duplicate an administrator who has no root OU, or whose root OU is outside yours.
- **Add administrator** and **Duplicate admin** follow the same rules. The new administrator never gets more than you may grant at each OU or device. Without **Change menu rights**, the new administrator gets no menu rights that you couldn't set yourself.
- If you sign in through several AD groups, the limit is your combined rights from these groups.
- If you save a rights template with rights that you don't hold, UniconOS Management Board saves the template without these rights and names them in a warning.

Because of these limits, UniconOS Management Board can refuse a change of rights that earlier versions accepted.

Upgrade

Existing administrators are granted all of the rights above, so they keep their current access after upgrading. Review and restrict them where you want narrower access, for example to set up a "now only" administrator (allow **Execute action immediately**, deny both **Schedule action** rights) or a "one device at a time" administrator (deny **Execute action on multiple devices**).

The **Software** rights and **Enrollment settings** are an exception. Until now, a right that a newer release added counted as allowed for administrators whose rights had been saved by an older release. For these rights, this is no longer the case. When you upgrade to UniconOS Management 2611, Scout converts the stored rights of every administrator:

- Administrators with **Edit administrators** keep every new right, so that they can grant these rights to others.
- All other administrators get the **Software** rights and **Enrollment settings** denied. Grant these rights to the administrators who need them.

The ELIAS menu rights of the legacy Scout Console don't carry over to the **Software** rights. After the upgrade, administrators may be refused actions that they could use before.

Scout Console

UniconOS Management Board uses the administrator records of the legacy Scout Console, which is end of life and not available for UniconOS Management 2611.

For the rights common to both the Console and UniconOS Management Board, see Administrator policy.

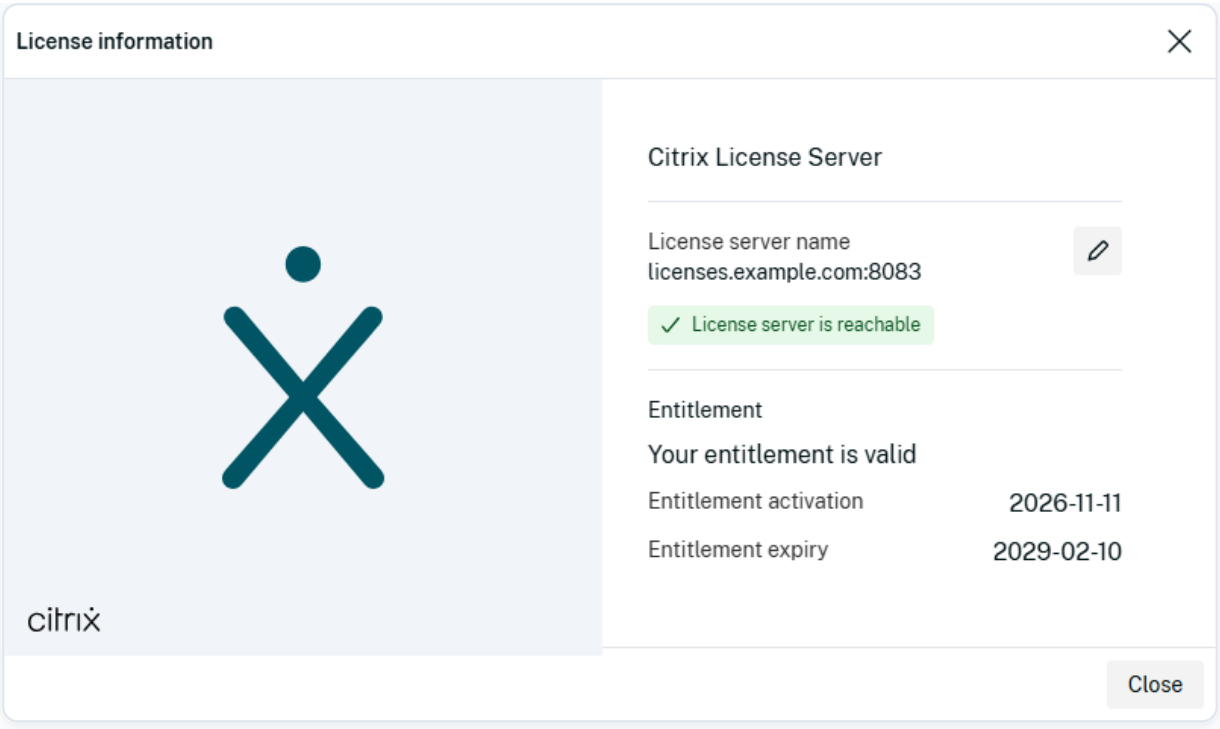
Licensing

Scout gets its license from a Citrix® License Server. The License Server holds your Scout licenses, and Scout Server checks with it regularly whether your entitlement is valid. In UniconOS Management Board, you see the result in the **License information** widget on the dashboard and in the **License information** dialog. To open the dialog, click the help icon (question mark) at the top right of UniconOS Management Board, and then click **License information**.

Scout needs Citrix License Server 2607 (build 56200) or later.

Messages in UniconOS Management Board name build 56100, the oldest build that Scout accepts. Use build 56200, which replaced it on the Citrix download page.

> **Important:**
>
> If you upgrade Scout from an earlier version and your License Server is older, upgrade the License Server first. Otherwise, Scout runs on its grace period after the upgrade, and devices can no longer connect to Scout Server when the grace period ends. For how setup checks the version of the License Server, see Upgrading to newer versions.



The dialog and the widget show the License Server and your entitlement:

- **License server name**: the configured License Server. UniconOS Management Board shows the name whether or not the server is currently reachable, together with the status of the server.
- **Entitlement**: **Your entitlement is valid** or **No valid entitlement has been found**, followed by **Entitlement activation** and **Entitlement expiry** dates. While a grace period runs, a **Grace period: N days remaining** line appears; click its information icon to open this dialog.

The status below the License Server name tells you whether Scout gets its license, and what to do if it doesn't:

STATUS	MEANING
License server is reachable	Scout Server reached the License Server at the last check.
License server is not reachable	Scout Server can't reach the License Server. Another line states when the connection last worked and when the last connection attempt was made. Check the configured name and port of the License Server.
License Server too old — upgrade to 2607 (build 56100) or later	The License Server is older than Scout needs. Upgrade the License Server, and then confirm its name again, see Configuring the License Server. Scout then gets its license without a restart.
License Server gives UniconOS Management no license — check its UniconOS Management licenses and its registration	Scout Server reaches the License Server, but the License Server holds no Scout licenses, or it isn't registered.
License Server certificate refused — configure the License Server again to accept its new certificate	Windows doesn't trust the certificate that the License Server presents, and it isn't the certificate that Scout accepted when the License Server was configured.

If UniconOS Management Board can't read the license state, the **License information** dialog shows **The license state is not known because the database layer did not answer. UniconOS Management checks again shortly.** and **System Status** shows **License state unknown**. UniconOS Management Board doesn't treat this as a missing license, and it checks again by itself.

Scout Server checks the License Server when the Scout Server service starts, every 12 hours, and right after the License Server configuration is saved. The dialog and the widget always show the result of the last check.

System Status, the status icon next to the help icon, shows the same information in two rows: one row for the License Server connection, one for the entitlement. A failing row shows as a warning, not an error.

Your devices get their licenses from Scout as well: every device gets the licenses for Fluendo Basic and Fluendo System. You don't need to assign or request these licenses.

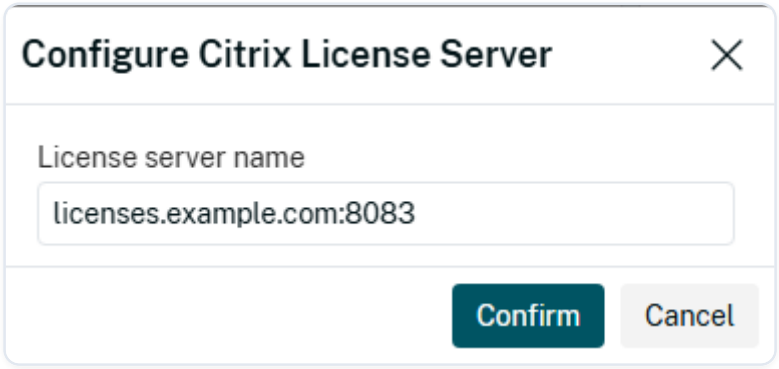
Scout also gives each device the end date of its software maintenance, which the device shows as **Subscription valid until**. With a License Server, this date is 12 months after the last license request of the device. A device requests its licenses when it starts, about every 7 days, and when Scout starts an update job for it. Devices refuse firmware that was built after this date. For the upgrade from an earlier version, see Upgrading Scout before the devices.

Configuring the License Server

You enter the License Server when you install Scout. To change it later, use the pen icon next to the License Server name, in the **License information** dialog or in the widget on the dashboard:

1. Click the pen icon.

   The **Configure Citrix License Server** dialog opens.

2. In **License server name**, enter the host name or IP address of the License Server and, if needed, its port. Leave out a protocol prefix such as `https://`, for example `licenses.example.com:8083`.
3. Click **Confirm**.

Scout Server then checks the License Server, and UniconOS Management Board shows the result:

- If the License Server is older than 2607, Scout keeps the previous configuration, and UniconOS Management Board shows **The License Server is too old. Upgrade it to License Server 2607 (build 56100) or later. The license configuration was not changed**.
- If Scout Server can't reach the License Server, Scout saves the configuration and starts its grace period. Scout Board shows **The license server is not reachable. The configuration has been saved. You are now in the grace period**.
- If the License Server holds no Scout licenses or isn't registered, Scout saves the configuration and starts its grace period. UniconOS Management Board shows **License Server gives UniconOS Management no license — check its UniconOS Management licenses and its registration**.

When you install Scout or configure the License Server in UniconOS Management Board, Scout remembers the certificate of the License Server. If the License Server later presents another certificate that Windows doesn't trust, Scout Server refuses it: the License Server counts as not reachable, Scout runs on its grace period, and UniconOS Management Board shows **License Server certificate refused — configure the License Server again to accept its new certificate**. A certificate that Windows trusts is always accepted.

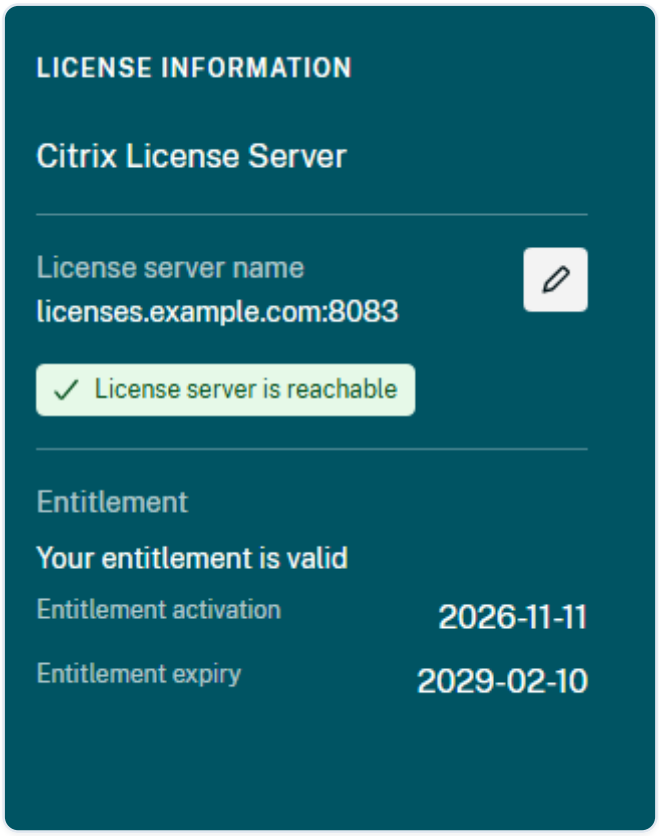
The certificate of the License Server changes, for example, when it's renewed or replaced, when the License Server is rebuilt or moved, or when a TLS proxy is put in front of it. To accept the new certificate, configure the License Server again in UniconOS Management Board. You don't need to restart Scout Server.

If several Scout Servers share the Scout database, UniconOS Management Board saves the new certificate on one of them only. On each other Scout Server, enter the thumbprint of the new certificate, 40 hexadecimal characters, as `LSCertHash` in the `[LS]` section of `eluxd.ini`, and restart the Scout Server service. `eluxd.log` names the thumbprint of the certificate that the License Server presents. Open `eluxd.ini` in an editor that runs as administrator.

To only log such a certificate instead of refusing it, set `LSCertCheck=0` in the `[LS]` section of `eluxd.ini`, and restart the Scout Server service. We don't recommend it. Any other value, and no value, refuses the certificate.

While Scout Server refuses such certificates, it doesn't follow a redirect from the address of the License Server. Enter the address at which the License Server answers itself.

The **License information** widget on the dashboard shows the same information as the dialog:



Operation and grace period

If Scout doesn't get a valid license from the License Server, Scout Server starts a grace period. This happens when the License Server isn't reachable, is too old, or gives Scout no license. During the grace period, Scout works as usual. Use this time to correct the License Server or its configuration.

The grace period lasts 14 days if Scout had a connection to the License Server and a valid entitlement within the last 5 days. Otherwise, it lasts 5 days.

When the grace period ends, devices can no longer connect to Scout Server, so you can't manage them. UniconOS Management Board then opens only a license page, where you can configure the License Server. Requests to the UniconOS Management Board REST API get HTTP status 402 until Scout has a valid license again. UniconOS Management Board shows the same page if Scout isn't licensed through a License Server at all. As soon as Scout has a valid license again, UniconOS Management Board and the devices work as usual.

Restoring or copying the Scout database

Scout keeps its license data in the Scout database. You can restore a backup of this database or copy it, for example to move Scout to another SQL Server or to test an update with a copy of your production database.

When Scout Server starts with the restored or copied database, it binds the license data to this database by itself and starts normally. You don't need to unlock the database with a validation code. Scout then checks your entitlement through the Citrix License Server as usual.

The stored passwords need the data key of the database. If you keep this key in a key file, copy the key file to the new server as well, see Data key for stored passwords.

> **Note:**
>
> Some installations use a license file instead of a License Server, by exception. Such a license file is bound to the database that it was issued for. It isn't valid for a restored or copied database, and it isn't meant for SQL Server AlwaysOn. If Scout doesn't accept a license file, UniconOS Management Board shows the reason and the database ID for which you request a new license file.

Additional information

For more information about the Citrix License Server and the licensing process, see:

- Citrix License Server
- Citrix License Manager

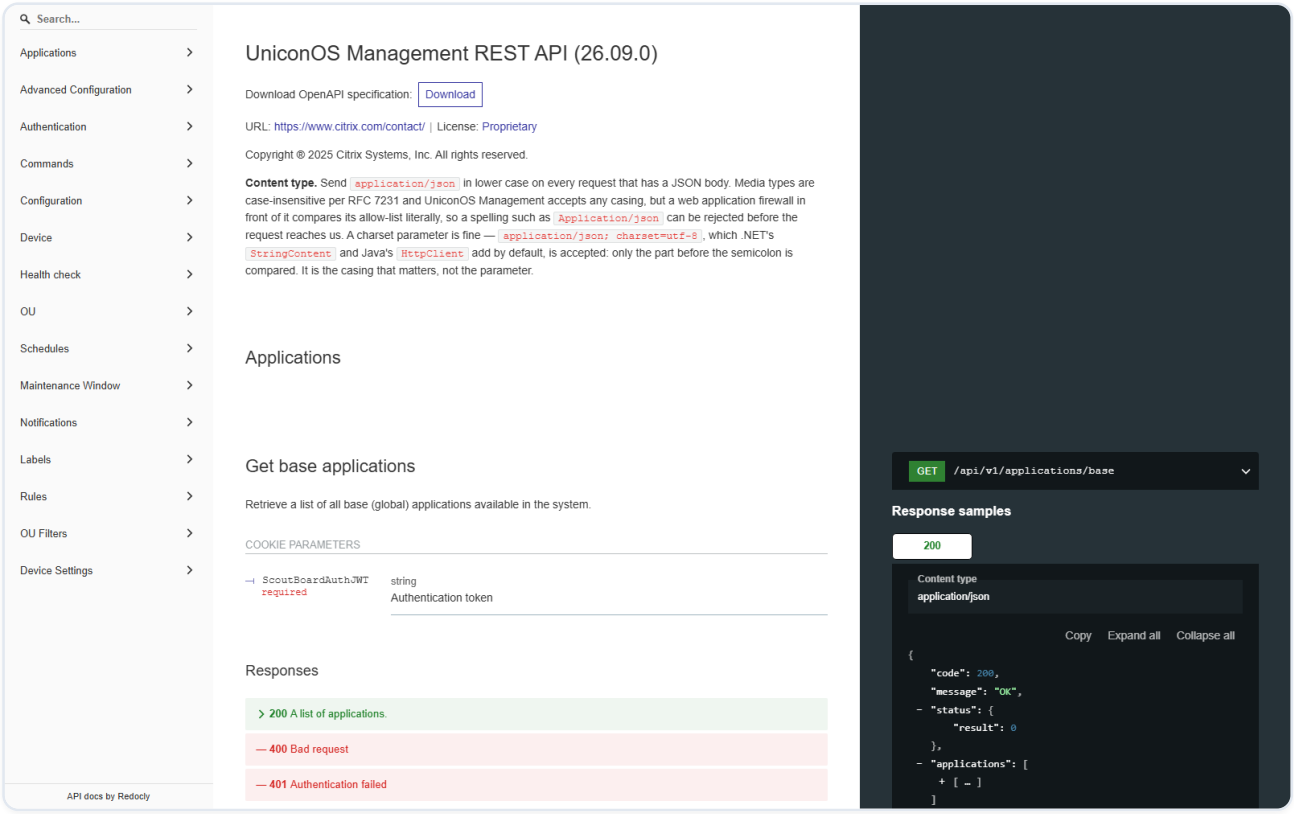
Using the REST API

UniconOS Management Board has a REST API, so that scripts and other applications can work with Scout without the UniconOS Management Board user interface. For example, a script can create OUs and devices, move devices to another OU or send commands to devices. You reach the REST API at the same address as UniconOS Management Board, and scripts sign in with UniconOS Management Board administrator accounts.

API reference

UniconOS Management Board includes a reference of all REST API endpoints, generated from its OpenAPI specification. It lists every endpoint with its parameters and possible answers, and many endpoints with examples. To open the reference, go to `https://<scout-board-host>:<port>/html/api-docs.html` in your browser. The address `https://<scout-board-host>:<port>/rest/api/v1/api-docs` leads to the same page. You don't need to sign in to read the reference, and it also works without internet access.

In the reference, the paths of the endpoints start with `/api/v1`. When you call an endpoint, put `/rest` in front of its path, for example `https://<scout-board-host>:<port>/rest/api/v1/devices`.



Signing in

A script signs in once with the user name and password of an administrator and gets a token. It then sends the token with every request. To sign in:

1. Write the user name and the password as JSON. Give the user name with its domain, either as `user@domain` or as `DOMAIN\user`. In JSON, the backslash must be doubled: `DOMAIN\\user`.

   ```
   {"username": "<user>@<domain>", "password": "<password>"}
   ```

2. Encode this JSON text in Base64.
3. Send the Base64 text as `loginData64` in a `POST` request to `/rest/auth/v1/login`:

   ```
   POST https://<scout-board-host>:<port>/rest/auth/v1/login

   Content-Type: application/json
   ```

```
{"loginData64": "<Base64 text>"}
```

The answer contains the token.

4. Send the token in the `Authorization` header of every further request:

```
Authorization: Bearer <token>
```

Note:

Base64 only encodes the user name and the password. It doesn't encrypt them. Connect to UniconOS Management Board over HTTPS, see Installing UniconOS Management Board.

Signing in with a user name and password needs Active Directory sign-in (`AUTH_TYPE=AD`). With another sign-in method, the request gets HTTP 404 with the message **AD authentication is not enabled!** Use the identity provider sign-in that is configured for this UniconOS Management Board instead. For the sign-in methods, see Configuring authentication options for UniconOS Management Board.

If the user name has no domain, or if UniconOS Management Board can't resolve the domain, the sign-in gets HTTP 412. To add a domain for REST API sign-in without showing it on the sign-in page of UniconOS Management Board, see Configuring AD domains for UniconOS Management Board.

Token lifetime

A token stays valid for 4 hours. UniconOS Management Board doesn't extend it while the script works, so when the token has expired, the script signs in again. To change the lifetime of all tokens, set `REST_TOKEN_LIFETIME` in the file `.env.custom` of UniconOS Management Board, for example `REST_TOKEN_LIFETIME=8h`, and restart UniconOS Management Board. If `JWT_EXPIRATION` is longer than 4 hours and `REST_TOKEN_LIFETIME` isn't set, tokens are valid for the time of `JWT_EXPIRATION`. A token is never valid longer than `SESSION_MAX_LIFETIME`, 12 hours by default.

To get a shorter lifetime, add `expirationTime` to the JSON of the sign-in, for example `"expirationTime": "30m"`. A longer lifetime isn't possible: UniconOS Management Board reduces it to the token lifetime. The value of `expirationTime` is a string with a positive whole

number and an optional unit: `m` for minutes, `h` for hours or `d` for days. A number without a unit counts as hours. Any other value gets HTTP 400.

Rights

The REST API checks the rights of the administrator account that signed in, in the same way as UniconOS Management Board does. A request that these rights don't allow gets HTTP 405. For example, exporting the OU structure through the REST API needs the object rights **Export** and **Export of the OU tree**, like **Export** in the OU menu of UniconOS Management Board. Importing an OU structure needs **Import** and **Import of an OU tree**.

A request that changes the configuration of an OU, a device or the base configuration changes only this configuration. If the request names an entry of another configuration, for example a printer, it gets HTTP 400, or HTTP 404 for a deletion, and nothing changes.

For the other rights that the REST API checks, see Administrator rights in UniconOS Management Board.

Answers and errors

Every answer has an HTTP status code, which tells you whether the request worked. Most answers are JSON, with the status code in `code`, a short text in `message` and the data in `response`. If a request fails, a `msg` field usually gives the reason, for example:

```
{"code":404,"message":"not OK","response":{"status":
{"result":3,"msg":"schedule not found"}}}
```

Not all errors have this form. Some answers to invalid input are plain text, for example `Error validating request body. "Name" is required.` So check the status code first, and read the body as text if it isn't JSON.

These are the status codes that you meet most often:

CODE	MEANING
200, 201	The request worked. Some requests that create an object answer 201.
400	The request is invalid, for example because a parameter or value is missing, unknown or wrong, or because an OU or a device field

CODE	MEANING
	doesn't exist. The message usually says what is wrong.
401	The request has no token. Sign in first.
402	The grace period of the License Server has ended, so Scout has no valid license. Requests work again as soon as Scout has a valid license, see Operation and grace period.
403	The request isn't allowed in this form, for example because the device password is missing or wrong.
404	The requested object, for example a schedule, a device or an application, doesn't exist.
405	The rights of the administrator account don't allow the request.
409	Someone else is editing the same configuration, in UniconOS Management Board or through the REST API. The message starts with `Someone else is currently editing this configuration`. Send the request again later.
412	Sign-in only: the user name has no domain, or UniconOS Management Board can't resolve the domain.
423	Scout has no supported license, for example because it has neither a License Server nor a valid license file, or because it rejected its license file. For a rejected license file, the message names the reason.
429	Too many failed attempts: too many failed sign-ins or too many wrong device passwords, see below.

The endpoints that look up or search devices accept in `properties` only the names of existing device fields. For any other name, the request gets HTTP 400, and the message names it. Earlier versions left such a name out of the answer.

The device and OU lookups compare the values that you send, for example a device name or an OU path, exactly as you send them. If your script escapes characters in these values itself, remove this escaping.

UniconOS Management Board limits failed attempts:

- After 10 failed sign-ins of one account from one address within 15 minutes, UniconOS Management Board refuses further sign-ins of this account from this address until 15 minutes have passed since the first of them. Successful sign-ins don't count. The limit applies to the sign-in page of UniconOS Management Board and to the REST API together.
- After 100 failed sign-ins of one account within an hour, from any addresses, UniconOS Management Board refuses sign-ins of this account until an hour has passed since the first of them.
- After 10 wrong device passwords of one administrator within 15 minutes, UniconOS Management Board refuses further requests of this administrator that need the device password, until 15 minutes have passed since the first of them.

Stored passwords

The REST API never returns a stored password, only whether a password is set: instead of the password, an answer contains `******`. This applies, for example, to Wi-Fi keys, application passwords and the passwords in configuration labels. Configuration exports don't contain stored passwords either. When you send a setting back, UniconOS Management Board handles its password fields like this:

- A password that you leave out, or send as `******` (six asterisks), stays as it is.
- A new password is stored encrypted.
- An empty password clears the stored password.

So a script can read a setting, change other values and send it back without touching the passwords. This also works for a printer and for LAN, WLAN and mobile internet profiles, which you can send back as you read them. The endpoints for configuration labels ignore a kiosk lock password: a label can't set it, and the kiosk lock password of the OU applies.

UniconOS Management 2609 and earlier returned stored passwords in an encrypted form instead of `******`. If a script sends back such a value, UniconOS Management Board recognizes it and keeps the stored password. A later version will save such a value as a new password, so change these scripts to send `******` or to leave the password out.

Scout encrypts stored passwords with the data key of the installation, see Data key for stored passwords.

Device password for commands

The device password protects commands that run with root rights on devices. It's the same for all devices that a Scout Server manages, see Local device password in the Scout guide.

When a script saves a user-defined command in the **UniconOS Command Scheduler** of the base configuration, an OU or a device, the request needs the device password in the `devicePassword` field. Without it, or with a wrong one, the request gets HTTP 403 with the message `A user-defined scheduler command requires a valid devicePassword`. Built-in commands and deleting a scheduler entry don't need the device password.

A custom command that a script sends to devices with root rights needs the device password too. For the field, see the API reference.

Activity log

Changes that a script makes through the REST API are written to the activity log, like changes in UniconOS Management Board, with the administrator account that made them. Requests that only read data, and refused requests, don't create entries.

Some entries start with `REST:`, for example `REST: Add label "<label name>"`. Other changes, for example commands for devices, appear in the same way as when you make them in UniconOS Management Board.

You find the activity log under **Security & permissions** > **Manage administrators** > **Activity log**. Scout writes an entry only if the **Monitoring level** includes the change. You set the monitoring level under **Security & permissions** > **Manage administrators** > **Settings**. For what each level records, see Monitoring levels in the Scout guide.

Loading the devices of an OU

`GET /api/v1/devices` returns the devices of an OU, and on request also the devices of all its sub-OUs. You choose which device fields the answer contains. This suits regular exports, for

example a nightly list of MAC addresses for network access control. In the API reference, you find the endpoint under **Device**.

The answer is JSON. If you need a CSV file, let your script convert the answer, as in the example below.

The request takes these query parameters:

PARAMETER	DESCRIPTION
`ouId` or `ouPath`	The OU, by its ID or by its path, for example `/<OU>/<sub-OU>`. Give exactly one of the two.
`includeSubOus`	`true` adds the devices of all sub-OUs. With `false`, the default, the list contains only the devices that are directly in the OU. Only `true` and `false` in lower case are valid.
`properties`	The device fields that the answer contains, separated by commas, for example `Name,Mac_Address,IP_Address`. Password fields can't be requested. Without `properties`, the answer contains the client identifier, name, MAC address, IP address, group ID and device ID.
`limit`	The number of devices per page, from 1 to 10,000. The default is 1,000.
`cursor`	The `nextCursor` value of the previous page. Leave it out for the first page.

Any other parameter, an invalid value or a device field that doesn't exist gets HTTP 400. For a device field that doesn't exist, the message names the field.

The list contains only the devices that the administrator account may see. An OU without devices gives an empty list. An OU that doesn't exist, or that the account may not see, gets HTTP 400.

Paging through the list

Each answer contains one page of devices in `response.devices` and a cursor in `response.nextCursor`. To get the next page, send the same request again and add `cursor=<nextCursor>`. When `nextCursor` is `null`, you have all devices.

Follow `nextCursor` until it is `null`, even if a page contains fewer devices than `limit`, or none at all. Pages can be shorter because the API leaves out the devices that the account may not see.

If devices are added or deleted while a script pages through the list, no other device is skipped or returned twice. A device that is added during the export may or may not be in the list.

This PowerShell example loads the names and MAC addresses of all devices of an OU and its sub-OUs, 10,000 per page, and saves them in a CSV file:

```
$headers = @{ Authorization = "Bearer <token>" }

$url = "https://<scout-board-host>:<port>/rest/api/v1/devices" +

    "?ouPath=<ou-
path>&includeSubOus=true&properties=Name,Mac_Address&limit=10000"

$devices = @()

$cursor = $null

do {

    $page = Invoke-RestMethod -Uri $(if ($cursor)
{ "$url&cursor=$cursor" } else { $url }) -Headers $headers

    $devices += $page.response.devices

    $cursor = $page.response.nextCursor

} while ($cursor)

$devices | Export-Csv devices.csv -NoTypeInformation
```

Reading and changing device settings

`GET /api/v1/settings/devices` returns the device settings that you find in UniconOS Management Board under **Preferences** > **Devices**, and `POST /api/v1/settings/devices` changes them. In the API reference, you find both under **Device Settings**. For what the settings do, see Preferences.

The settings come as one JSON document with five parts:

PART	SETTINGS
`ouFilter`	Whether Scout assigns new devices to an OU by OU filter, and with which type of filter
`newDevices`	The default OU and the other options for new devices
`deviceNames`	The template that Scout uses to name devices, and its options
`discover`	The response time for device discovery, in milliseconds
`retainLocalConfig`	Whether devices keep the values that users set in unlocked fields

In `ouFilter`, `mode` is `disabled`, `subnet` or `custom`. `disabled` turns off the OU filter, `subnet` stands for the **Subnet filter** and `custom` for the **User-defined filter**. Rule-based OU filters take effect only when `mode` is `custom`. To manage the OU filters themselves, use the endpoints under `/api/v1/ouFilters`.

With `POST`, send only the parts and fields that you want to change. All other settings stay as they are, and the answer contains the complete document. For example, this request sets the response time to 500 milliseconds and turns off retaining the local configuration:

```
POST https://<scout-board-host>:<port>/rest/api/v1/settings/devices

Content-Type: application/json

Authorization: Bearer <token>



{"discover": {"responseTimeMs": 500}, "retainLocalConfig": false}
```

Send values with their JSON type, for example `false` and not `"false"`. A field that doesn't exist, a value of the wrong type, an invalid value or an empty request gets HTTP 400, and no setting changes. A change that the API accepts applies in the same way as a change that you save in UniconOS Management Board.

Both requests need the menu right **Devices** under **Menu : Options -> Advanced options**. Each successful change creates one entry in the activity log that names the changed parts, for example `REST: Device settings are saved (ouFilter, deviceNames, discover)`.

Monitoring with the health check

`GET /api/v1/healthcheck` reports the state of Scout for monitoring tools, for example whether the Database Layer answers and whether Scout has a valid license. Like every other request, it needs a token, see Signing in.

To get the license state, add the parameters `license` and `subscription`:

```
GET https://<scout-board-host>:<port>/rest/api/v1/healthcheck?
license&subscription

Authorization: Bearer <token>
```

The answer then contains the sections `license` and `subscription`. In each section, `result` and `msg` show the license state:

RESULT	MSG	MEANING
`0`	`valid`	Scout has a valid license. With a license file, the `license` section also shows the number of device licenses in `totalLicenses`, `usedLicenses` and `availableLicenses`.
`1`	`not valid`	Scout has no valid license, for example because the grace period of the License Server has ended, or because Scout rejected its license file. For a rejected license file, `reason` names the cause, for example `other-database` for a file that was issued for another database.

RESULT	MSG	MEANING
3	license state unknown	UniconOS Management Board can't read the license state, for example because the Database Layer doesn't answer.

From UniconOS Management 2611, the health check answers with HTTP status 200 also when Scout has no valid license. Earlier versions answered with HTTP status 423 or 402, as other requests still do, see Answers and errors. If your monitoring evaluates these status codes, evaluate the `license` section instead.

If several Scout Servers share the Scout database, add the parameter `scoutNodes` to check the connections between the servers, see Checking the health of the servers.

Scout Cloud Gateway

Scout Cloud Gateway is a solution for managing devices that are on the Internet and have to be connected to their Scout Server via VPN.

Overview

Unicon's Scout Cloud Gateway is a convenient solution for managing devices that are on the Internet and have to be connected to their Scout Server via VPN. Home office devices or mobile devices in changing environments can thus be conveniently and securely connected to the existing infrastructure without the need for a VPN backend.

The Scout Cloud Gateway (SCG) comes with a **VPN infrastructure** - the VPN connection is established and provided automatically. The first time an eLux® device connects, users are required to authenticate. After successful authentication, the SCG transmits the VPN configuration and Scout Server address to the device. This communication uses the secure management protocol.

eLux RP 6 devices have the VPN software already on board.

For each device, the system creates a certificate for client authentication which the administrator can view, revoke or renew at any time.

The Scout Cloud Gateway is tailored to eLux device management via the Scout Enterprise Management Suite and supports environments with multiple Scout Servers. One gateway instance can connect around 1024 devices from the Internet. To connect a larger number of devices, you can install **multiple gateway instances** in a cluster. The instances exchange necessary information with each other. Clustering makes the Scout Cloud Gateway **scalable**: Thousands of devices in the home office can be connected smoothly.

A further advantage is that the devices can be used **dynamically both from the Internet and the Intranet**. With the appropriate network configuration, the devices can be easily integrated and operated in their environment via the Internet or Intranet. The system then automatically recognizes whether it is on the Internet or the Intranet. Only when used outside the Intranet for the first time, users must enter their logon data for the Scout Cloud Gateway.

This convenient option for dynamic use can also be used for preparing and rolling out new devices.

The Scout Cloud Gateway is offered as a Debian installer and as a virtual machine with an image using Ubuntu 24.04. For further information, see System requirements.

To configure and manage your gateway, the SCG WebAdmin interface is provided.

Access control

To connect new devices, the Scout Cloud Gateway (SCG) needs to ensure that the user is authenticated. To do this, a connection to Active Directory or to the identity management system Keycloak can be set up. Keycloak allows user authentication strictly separated from the SCG. As a third variant, you can define tokens for access directly at the SCG.

After a user has been successfully authenticated, a certificate is issued for the corresponding device, which is checked against at the next logon. As long as the certificate is valid, no new user logon to the SCG is required. The RSA certificates have a key length of 4k. Administrators may configure the renewal of certificates and get a quick overview of expiring certificates in the Devices view and the Dashboard 2 of the WebAdmin interface.

User authentication via Active Directory

Active Directory handles the authentication and the assignment of the device to its OU. To control the OU assignments, the AD environment must be configured in advance.

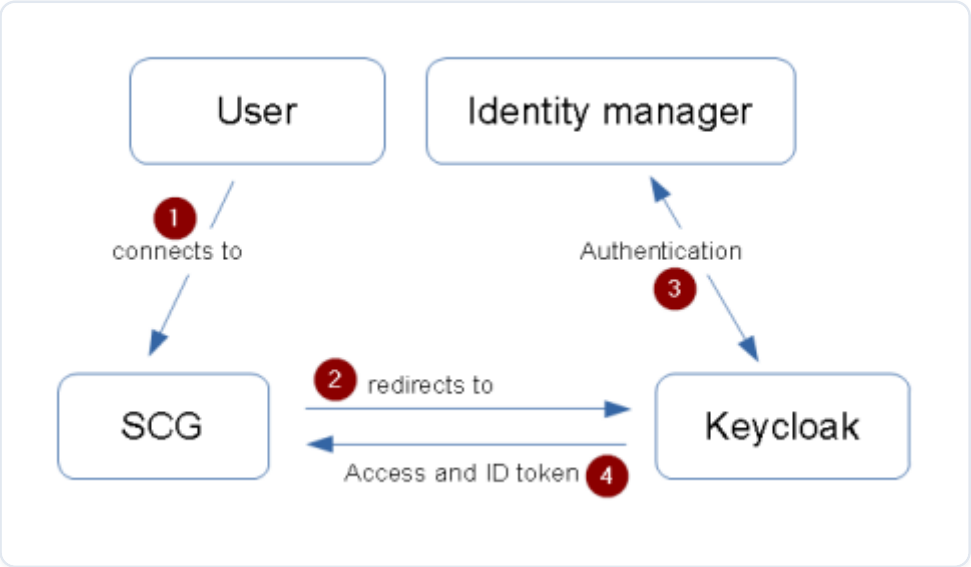
Authorized AD users then log on to the Scout Cloud Gateway once with their device. The AD logon data are evaluated by the AD server. Next to user authentication the devices are registered on the Scout Server in the OU specified by the user assignments.

In the **Devices** view of the WebAdmin interface, usernames are displayed with their domain.

User authentication via Keycloak

Keycloak is an implementation of the OpenID Connect protocol and can be used perfectly for connecting devices via the Scout Cloud Gateway. The user authentication is processed completely separated from the SCG and thus protects the logon data. Also, the Keycloak server must use signed certificates for TLS communication.

Keycloak also acts as an identity broker and supports user accounts from other identity and access management systems, including Active Directory.



The users are forwarded via the Scout Cloud Gateway (1) to Keycloak for logon (2). In the logon dialog of the identity manager defined in Keycloak, they enter their logon data, which are then checked by the defined identity manager (3). After successful authentication, Keycloak returns an access and ID token for the user to the SCG (4). This procedure completely isolates the SCG from the users' logon data.

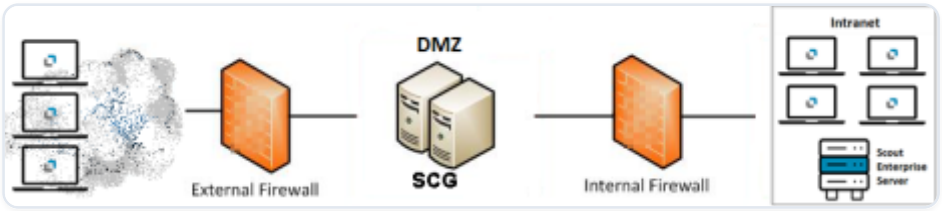
Users who are authenticated via Keycloak are always assigned to the default OU that is set in UniconOS Management Board under **Preferences** > **Devices**, see Devices in the **UniconOS Management Board** guide. In the **Devices** view of the WebAdmin interface, their usernames are displayed without a domain.

User authentication directly via Scout Cloud Gateway

Users receive a token from the administrator to register their device. The tokens are defined after installation in the SCG WebAdmin interface. To each token, you assign an OU in which the user's device is to be registered.

Scout Cloud Gateway in a DMZ

For the setup of the Scout Cloud Gateway in your network, we recommend the operation in a security-controlled 'Demilitarized Zone',



The following routes must be set up for configuration:

- From the eLux® devices to the DMZ

  You define these routes in the SCG WebAdmin interface under **Configuration > VPN device settings > Routes**.

- From the Scout Cloud Gateway to the Intranet

  For further information, see Static IP routes.

- From the Scout Server, AD server / identity manager and other required (application) servers back to the Scout Cloud Gateway in the DMZ

  For further information, see General network settings.

Use dedicated firewall rules to control access from the Internet to the DMZ and from the DMZ to the Intranet.

> **Note:**

> To use the Scout Cloud Gateway as a router between DMZ and Intranet, you can use a second network card. In the VM installation, two network cards are already considered and can be configured.

> Note the following

- If you intend to use multiple SCG instances in a cluster, they must also be able to communicate with each other.
- If you define a gateway and DNS only for the internal network, make sure that the names of the Scout Server and AD server / identity manager can be resolved.

Example for configuring routes

Premises:

(A) eLux devices are supposed to access resources on the Intranet (network 192.168.50.0/24).

(B) The SCG is located in a DMZ with network 192.168.100.0/24 and connects eLux devices via tunnel network 10.8.0.0/24.

(C) Both networks are connected via a router/gateway, with address 192.168.100.1 in the DMZ and 192.168.50.1 in the Intranet.

Required routes:

A) Network: 192.168.50.0, subnet mask: 255.255.255.0 (gateway is the VPN server of the SCG)

B) Network: 192.168.50.0, subnet mask: 255.255.255.0, gateway from DMZ to intranet: 192.168.100.1

C) Network: 10.8.0.0, subnet mask: 255.255.255.0, gateway from the intranet into the DMZ: 192.168.50.1

For further information, see General configuration.

System requirements

General requirements

- System resources
  - 4 CPUs
  - 8 GB RAM
  - 24 GB hard disk space
- Network
  - One IP address in the VPN address range for each connected eLux® device
  - Internal HTTPS access to the SCG WebAdmin interface (Interface for configuration and management)
  - Bi-directional connection (via network card or firewall rules) of eLux devices to the Scout Server and of the Scout Cloud Gateway to the Identity provider
- Scalability by means of clustering

If required, multiple machines can be installed as SCG servers. 1024 devices can be connected per server.

You can set up your instance with the help of a Debian Installer under a Linux system. Alternatively, a ready configured virtual appliance is available. In both cases, the network must be configured before the SCG is started to enable routing from the DMZ to the internal network. The virtual machine assists with the setup.

	DEBIAN PACKAGE	VIRTUAL MACHINE
Download	Debian package (.deb)	Virtual appliance (.ova)
		The virtual machine is installed using an .ova template (Open Virtual Appliance).
Required system	Configured Debian Linux system	Virtualization platform such as VMware
	We recommend Ubuntu 24.04.	
Specifics		Wizard supports system configuration
		Additional functions in the WebAdmin interface
		Upload public keys for SSH access to the instance
		Upload individual HTTPS certificates for the WebAdmin interface connection (to replace the automatically created self-signed certificate)

Availability from the Internet

- **Management protocol** between eLux devices and Scout Server

For connecting the devices, the SCG server should be reachable from the Internet as FQDN. The default port for communication is **22125**.

If you are running multiple SCG instances in a cluster, we recommend that you distribute the requests to the different instances via router.

- **VPN port for connecting devices**

For cluster operation: Each SCG instance requires a unique combination of FQDN and port for its VPN server.

The administrator performs the configuration in the SCG WebAdmin interface, see Gateways.

Internal availability

- **HTTPS Port 443** for connecting the WebAdmin interface (configuration, clustering, management)
- If Active Directory or Keycloak is used, the servers must be reachable
- Only virtual machines (.ova): **port 22 for access via SSH** to the instance for administrative access (optional)
- For cluster operation: **MongoDB port 27017** for internal communication of the databases across the SCG instances

> **Note:**

> Make sure that your Linux system is always up to date. To do so, use the standard commands **apt-get update** and **apt-get upgrade**.

> For support periods and the compatibility matrix see the Whitepaper **Releases, Lifecycles and Compatibility**.

Installation

The Scout Cloud Gateway (SCG) may either be installed and run as a Debian package on Linux or operated on a virtual machine (VM).

For the **VM** installation, you need a virtualization platform. Unicon™ provides an OVF template (.ova) that you use to install your VM. The MongoDB database and web server used by the Scout Cloud Gateway are installed and preconfigured with it.

The **Debian** variant is installed with a Debian package. You install the MongoDB database yourself, the configuration for database and web server is supplied.

To configure your SCG instance, connect to it via a web browser and use the SCG WebAdmin interface.

To give users and their devices access to the corresponding OUs, define AD groups with a defined prefix and the OU ID in your Active Directory system. Then assign the users to them. Alternatively, configure Keycloak for authentication or create SCG accounts. For further information, see General configuration.

The devices will be integrated via the eLux® First Configuration Wizard.

Scout Cloud Gateway as a Debian machine

The Scout Cloud Gateway may be operated in a Linux environment. A Debian package (.deb) (tested with Ubuntu 24.04) is available for this purpose.

The Scout Cloud Gateway works with a **MongoDB** database. As a web server, you can use **NGINX** or **Apache**.

> **Note:**

> Clusters are supported also for a Debian installation.

> The connection between WebAdmin interface and web server is always made via HTTPS. The certificate for this is automatically created during installation.

Networking requirements

For the routing to work, please make sure that IP forwarding is enabled. The value for `net.ipv4.ip_forward` needs to be set to `1`, using the following command.

```
sysctl net.ipv4.ip_forward=1 -w
```

To ensure the IP forwarding setting persists after a reboot, create the file `99-scg-ip-forwarding.conf` in the folder `/etc/sysctl.d/` with the following content:

```
net.ipv4.ip_forward=1
```

Debian package dependencies

A Debian package is available for installation in a Linux environment.

- **MongoDB** version 8

  The MongoDB database is preconfigured.
- Web server **NGINX** or **Apache 2.x** Configuration data for both variants is supplied and must be activated after installing the SCG Debian package. In the example, we will use NGINX.
- **Node.js** version 24.22.1 or a later 24.x version
- **OpenVPN** for the VPN tunnel to the eLux® devices
- **jq** json tool, **moreutils**, **logrotate** and **zip** to install the Debian package.

Installing the tools and database

> **Important:**

> The SCG machine's hostname may not be changed after the MongoDB database has been installed and configured. If required, change the hostname before.

> The following instructions use Ubuntu 24.04 and NGINX as a web server.

1. Add the package sources for MongoDB.

   ```
   wget -qO - https://www.mongodb.org/static/pgp/server-8.0.asc | gpg --
   dearmor -o /usr/share/keyrings/mongodb-server-8.0.gpg



   echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-
   server-8.0.gpg ] https://repo.mongodb.org/apt/ubuntu noble/mongodb-org/
   8.0 multiverse" | \

   tee /etc/apt/sources.list.d/mongodb-org-8.0.list
   ```

2. Add the package sources for Node.js.

```
wget -qO - https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key |
gpg --dearmor -o /usr/share/keyrings/nodesource.gpg



echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/
nodesource.gpg ] https://deb.nodesource.com/node_24.x nodistro main"  |
\

tee /etc/apt/sources.list.d/nodesource.list > /dev/null



echo -e "Package: nodejs\nPin: origin deb.nodesource.com\nPin-Priority:
600" > /etc/apt/preferences.d/nodejs
```

3. Install MongoDB, Node.js and further dependencies.

```
apt-get -y update



apt-get -y install mongodb-org nodejs nginx openvpn jq moreutils zip
logrotate
```

4. Check for the MongoDB version. The version must be **8**.

```
mongo --version
```

5. Check for the Node.js version. The version must be **24.22.1** or a later 24.x version.

```
node -v
```

Note:

You can install the SCG Debian package now.

Installing the Debian package and configuring the web server

REQUIRES
Node.js and further tools are installed. The MongoDB database is installed and configured.

1. From our Citrix Downloads portal, download the SCG Debian package `scg_<version>_amd64.deb`.
2. Install the Debian package by using the following command:

   ```
   dpkg -i scg_< Version>_amd64.deb
   ```

   a. Start the SCG service.

   ```
   systemctl start scg-controller
   ```

   b. Optionally, configure the components for automatic system start-up.

   ```
   systemctl enable  'nginx' 'cron' 'scg-controller'
   ```

3. Configure the NGINX web server:

   By default, the SCG installs a functional web server configuration under the path `/etc/nginx/conf.d/scg.conf`. Here, self-signed certificates are linked for initial communication.

   a. Optional: Replace the self-signed certificates with your own. To do so, edit the file `/etc/nginx/conf.d/scg.conf` and replace the certificate files

   ```
   ssl_certificate /opt/unicon/scg/certs/http/certificate.crt;

   ssl_certificate_key /opt/unicon/scg/certs/http/certificate.key;
   ```

   b. Make sure that the SCG configuration is not overwritten by the NGINX default configuration.

   ```
   rm /etc/nginx/sites-enabled/default
   ```

   c. To customize the NGINX user, edit the file `/etc/nginx/nginx.conf` (from SCG 1 2209)

```
user scg-web scg-web;
```

d. Restart the service.

```
systemctl restart nginx
```

The installation is complete. In the next step, use a browser to log on to the SCG WebAdmin interface and configure your SCG instance. For further information, see Gateway configuration in the SCG WebAdmin interface.

- After you have configured your SCG in the SCG WebAdmin Interface, to check whether all components are running, use the following command:

```
systemctl status --no-pager --lines=0 'scg-mongod' 'nginx' 'cron' 'scg-openvpn' 'scg-backend' 'scg-controller'
```

Uninstalling the Debian package

To uninstall the Debian package, use the following command:

```
apt-get remove scg
```

To additionally delete the configuration data of the last installation, append the `--purge` parameter.

Scout Cloud Gateway as a VM

The Scout Cloud Gateway is provided as a virtual machine. Based on the OVF template (.ova), you install a virtual machine that you can vary the configuration of.

The following WebAdmin interface features are supported in VM installations:(but not in a Debian installation)

- Upload public keys for SSH access to the instance
- Upload individual HTTPS certificates for the WebAdmin interface connection (to replace the automatically created self-signed certificate)

Installing the VM

REQUIRES
Virtualization platform and computing resources (4 CPUs, 8 GB RAM, 24 GB hard disk) for installing virtual machines based on an [.ova] template (Open Virtual Appliance)

1. From our Citrix Downloads portal, download the current `.ova` template.
2. Within your virtualization platform, install a virtual machine from the template `scg-<version>.ova`, for example by selecting **Deploy OVF Template**.
3. Optionally, assign a host name for the new MAC address via DHCP. Otherwise, specify a name in the next step on the machine along with first network settings.
4. Power on your machine.

A wizard will guide you through the initial VM configuration; setting the root password, the FQDN and network settings.

Setting first configuration settings on VM

Configure the root password, host name and network cards directly on the machine.

To operate the Scout Cloud Gateway in a DMZ, you may configure two network interfaces for internal and external LAN. For further information, see Scout Cloud Gateway in a DMZ.

1. Start a remote console of the VM.

   > **Note:**
   >
   > The English keyboard language is active.

2. Use the following keys for navigation:

KEY	DESCRIPTION
`Tab`	Jump to next element
`Space`	Enable or disable selected option
`Return`	Trigger selected button

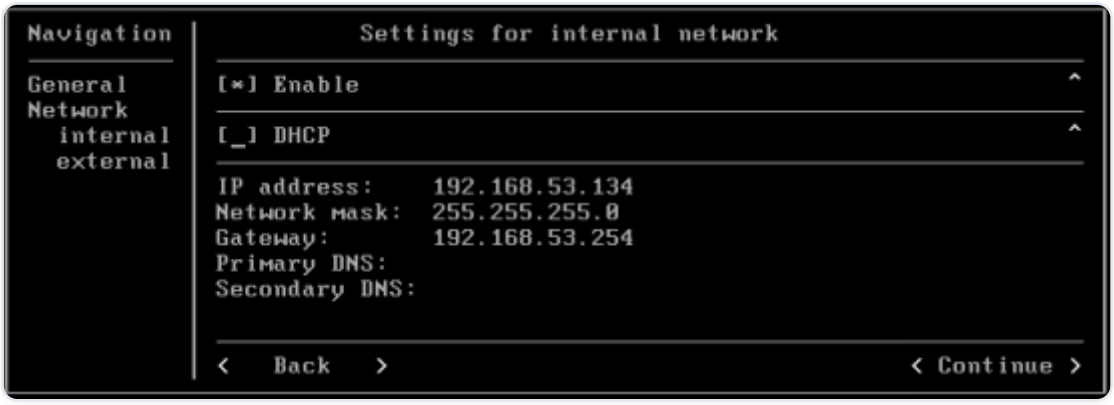
3. Under **General**, set and confirm an individual root password.
4. Under **Network**, optionally change the host name of your SCG machine.

   If you enter the host name as FQDN, avoid using `.local` as the domain.

   > **Important**
   >
   > The host name must not be changed later on.

5. Under **Network**, specify your NTP server.
6. Configure the internal and external network settings. Under **internal** and **external**, define whether the respective network card is used and whether DHCP is to be used. If you disable DHCP, you can configure the IP address and other data yourself.

   The first network card is defined as **internal**, the second as **external**.

   For further information on two network interfaces in the DMZ, see Scout Cloud Gateway in a DMZ.

Then, use a browser to log on to the SCG WebAdmin interface.

Connecting and logging on to the SCG instance

To manage your Scout Cloud Gateway, the WebAdmin interface is provided.

> **Note:**
>
> The system uses a secure HTTPS connection to the web server, for which a self-signed certificate was created during installation. We recommend that you replace the self-signed certificate with your own certificate.

1. In your browser, connect to your machine via HTTPS. For the URL, you can use the host name or the IP address. Example: `<https://192.168.54.24>`
2. Log on to the SCG WebAdmin interface with the username admin and initial password elux.
3. Enter a new password.
4. In the next step, in the **Configuration** view, configure your gateway.

The SCG WebAdmin interface is initially displayed in the language configured in the browser settings (German or English). For further information, see Logon and language.

In addition to the **Configuration** view, the WebAdmin interface offers the **Dashboard**, **Gateways** and **Devices** views.

Upgrading to later versions

> **Note:**
>
> We recommend that you make a backup before you upgrade.

Via installation file

(for Debian and .ova installations)

> **Note:**
>
> SCG supports an in-place update of the SCG components, if the update only involves the SCG components.
>
> With the SCG Version 1 2605, the underlying system has been updated; including OS (Ubuntu version 24.04, MongoDB 8, and all other system components). This is a big system-wide change. As an in-place update of system components is not possible, a migration is needed by creating new VM instances and migrating the data. To achieve this, SCG version 1 2605 provides scripts and UI based elements to export and import the data. If you are updating to SCG version 1 2605, please continue with the **Migration Guide** below.

In-place update, for updated SCG components

1. From our Citrix Downloads portal, download the current SCG Debian package `scg_<version>_amd64.deb`.
2. Copy the file to your SCG instance and connect to it via SSH.
3. Update Node.js to version 24.22.1 or later. From SCG 1 2605.1 on, the SCG package can't be installed with an older Node.js version.

```
apt-get -y update



apt-get -y install nodejs
```

4. Enter the following command:

```
dpkg -i scg_< Version>_amd64.deb
```

> **Note:**
>
> During installation, the package manager checks for configuration changes. This is the case for the NGINX web server configuration. Apply the maintainer version and add any custom changes manually.

5. Only Debian: Check the web server configuration and restart the web server. For further information, see Installing the Debian package and configuring the web server.
6. If you have multiple instances, update all instances one by one using the above procedure.
7. Connect to an instance via WebAdmin interface and check whether the gateway configuration has been correctly applied.
8. Also, reset the browser cache to update outdated data. For example, use the key combination CTRL+F5 (Firefox).

Migration guide to SCG version 1 2605 from SCG version 1 2409

This guide describes the update process from a SCG version 1.2409 to 1.2605.

The new version has the most up-to-date system components and the upgrade needs to be done by creating a new SCG instance (or cluster) and migrating the data from the previous instance (or cluster).

Scripts are being provided for the migration, which will export the data from the previous version and import it to the new version.

> **Note:**

> In the SCG administration application, on the top-right burger menu, you can import the data from an SCG instance with the previous version (1 2409) directly. Alternatively, or if the UI based import does not work, you can use data export and import scripts.

> The new version provides this script in `/opt/unicon/scg/scripts/dataExport.sh`

This script needs to be copied to the old instance (**must be the primary instance**) and made executable

```
chmod +x dataExport.sh
```

```
./dataExport.sh
```

Running the script will export the data to `/tmp/` as a JSON file.

> **Note:**

> This JSON file can be imported in the new SCG version using the import dialog, which can be accessed via the burger menu on the top-right of the SCG administration application. You can also use the dataImport.sh script in the new SCG instance.

Via cluster process

> **Important:**

> The SCG version 1 2605 comes with an updated system and new MongoDB version. A direct clustering approach for updating from a pre 2605 version needs to be performed using export (from the previous version) and import (to the new version 2605) of the data!

> If you are updating to SCG version 1 2605, please continue with the **Migration Guide** above.

> Initial situation: You have an existing Scout Cloud Gateway installation with minimum one instance in use. The registered devices are connected to the SCG server via a DNS entry. When updating to a later SCG version, you want the devices to remain on-boarded and configured as they were before.

To upgrade the version of an SCG installation (A) to a later version (B), create a new instance with version B and then transfer the configuration from A to B using the cluster function. To do so, perform the following steps.

1. Install an additional virtual machine (.ova) with the new Scout Cloud Gateway version B, see Installing the VM.
2. On the machine, configure the hostname and networks, see Setting first configuration settings on VM.
3. Log on to the WebAdmin interface of the new instance B via a browser, see Connecting and logging on to the SCG instance.

   > **Note:**
   >
   > Please do not perform any configuration of instance B. The configuration is automatically applied through the synchronization of the cluster. Instance-specific configuration is done after clustering.

4. From instance B, add the old instance to the cluster:
5. Wait, until the new instance initializes
6. Configure the gateway settings for instance B. To do so, select instance B from the list box under **Configuration > Gateways** and configure the VPN connections. For further information, see Gateways.

   *Both gateway instances are now shown with their properties in the **Gateways** view.*
7. In the **Gateways** view, deactivate instance A, see Gateways view.
8. Shut down the virtual machine A.
9. Set your DNS entry to the IP/host name of instance B.
10. Verify whether instance B allows for the client connections. To do so, you may use the following options:

- The **Gateways** view must show the public host name and port of instance B.
- In the WebAdmin interface of instance B, perform a health check. All components must be up and running.
- On-board a new device or restart a connected device. The device must be assigned to gateway B in the **Gateways** view.

11. In the Gateways view, delete instance A.

Instance B has now adopted the entire configuration of instance A, including the devices. As soon as a device restarts, it connects to the new instance B.

> **Note:**
>
> Also use this procedure if you want to change the host name of a gateway instance (without updating to a later version).

General configuration

Configuring your Scout Cloud Gateway consists of various tasks such as preparing user authentication or setting general network settings.

Subsequently, configure your SCG instance in the SCG WebAdmin interface.

Configuring Active Directory environment

All users whose devices are to be connected to their Scout Server via the Scout Cloud Gateway must be authorized via AD groups and assigned to an OU.

1. In your AD system, create a group for each OU of a Scout Server that you want users to access through the gateway.

   The group names must have the following format:

   Prefix that is defined in the SCG WebAdmin interface (`SCG_Group_`) and the ID number of the OU (`42`)

   Example: `SCG_Group_42`

   The prefix identifies groups for the Scout Cloud Gateway.The devices used by the members of the created group to log on are automatically assigned to the defined OU ID. The prefix is

predefined as SCG_Group_ and can be changed in the gateway configuration of the SCG WebAdmin interface.

UniconOS Management Board shows the ID of an OU as **OU ID** in the details of the OU, see OU details.

If the OU is protected by a password or hidden, new devices enter the default OU instead, unless the setting **Devices that join a protected or hidden OU without sending its password to Scout Server** is set to **Accept and log**, see Devices in the **UniconOS Management Board** guide.

2. Add the relevant AD users to the defined groups.

> **Note:**
>
> Avoid assigning a user to multiple groups with this prefix, since a device can only be assigned to one OU at a time.

Users are authorized to access their Scout Server via the Scout Cloud Gateway. They can register with their devices as soon as the gateway is configured. The devices will be assigned to an OU via the AD users and their membership in a specific AD group.

Configuring Keycloak

> **Note:**
>
> Keycloak support is deprecated. Please do not use it anymore.

> Keycloak is an open source solution for identity and access management and is based on the OpenID Connect protocol. In order to use Keycloak for user logon, Keycloak must be configured on the server side and a Keycloak client must be installed.

All users whose devices are to be connected to their Scout Server via the Scout Cloud Gateway must be authorized via Keycloak roles.

1. Set up a Keycloak server.

> **Note:**

> To protect sensitive data, only HTTPS configuration is supported. For this you need a CA certificate.

2. In the Keycloak administration console, create a realm for your environment.
3. Create a Keycloak client for your application. For the Keycloak client, edit any options labeled on the figure below.

OPTION	DESCRIPTION
Settings > Client ID	Name of your client / application
Settings > Valid Redirect URIs	One or more SCG addresses to which Keycloak forwards users after successful authentication

4. Within your client, create roles and users. Assign a role to each user.

*Users are authorized to access their Scout Server via the Scout Cloud Gateway and can register their devices once the gateway is configured. The devices are assigned to the default OU that is set under **Preferences** > **Devices** in UniconOS Management Board.*

General network settings

Routes in the network

In the relevant environment, configure the destination IP address and address range of the following connections to the eLux® devices you want to route via the secure communication of the Scout Cloud Gateway:

- Scout Server
- Computers on which you mirror devices in UniconOS Management Board (WebRTC, port 5902)
- Web server (images and eLux software packages)
- DNS server

Example for one instance: `route -p add 10.8.0.0/24 192.168.16.5`

For clusters, you need more routes accordingly.

Dynamic use of devices in the Internet and intranet

If you want to connect mobile devices both via Internet and intranet, configure the same FQDN for the Scout Server and the Scout Cloud Gateway. As a result, the FQDN in the intranet is resolved directly to the Scout Server. If the eLux device is used in the Internet, the FQDN will be resolved to the Scout Cloud Gateway. For further information, see DHCP configuration.

On each restart, the device checks whether the IP address of the resolved FQDN is a Scout Server or a Scout Cloud Gateway.

> **Notes:**
>
> - If you use different FQDNs for the Scout Server and Scout Cloud Gateway, note that each time the operating environment changes between the intranet and Internet, a factory reset of the eLux device is required.
> - For VPN connections, a separate web server for firmware updates can be configured in the Scout device configuration.

Integrated VPN

For the Scout Cloud Gateway, OpenVPN is operated in routing mode (TUN device). This operating mode is recommended as the more efficient method: The VPN operates in layer 3 carrying only IP packets, which reduces bandwidth load.

For each connected device, one IP address is required (subnet topology).

For a given number of devices to be managed, determine the required subnet mask that filters your IP address (VPN range). In **Configuration > Gateways**, for your instance, enter the subnet mask in dotted decimal notation.

Examples:

SUBNET MASK/CIDR	SUBNET MASK/DOT-DECIMAL	MAX. NUMBER IP ADDRESSES
`/27`	`255.255.255.224`	`30`
`/26`	`255.255.255.192`	`62`

SUBNET MASK/CIDR	SUBNET MASK/DOT-DECIMAL	MAX. NUMBER IP ADDRESSES
`/25`	`255.255.255.128`	`126`
`/24`	`255.255.255.0`	`254`
`/23`	`255.255.254.0`	`510`
`/22`	`255.255.252.0`	`1022`

The maximum number of available IP addresses corresponds to the maximum number of devices that can be connected.

The device manages the leases itself and acts like a miniature DHCP server. One lease is valid for 10 minutes.

Static IP routes

To ensure the accessibility of your resources such as Scout Server, Active Directory, NTP and DNS, add static IP routes to the SCG machine. Static routes can be set for the external and internal network interface on the instance in the relevant configuration file provided:

`/etc/systemd/network/00-external.network`

`/etc/systemd/network/00-internal.network`

The syntax to be used can be found in the freedesktop documentation for the **systemd-networkd** service `<https://www.freedesktop.org/software/systemd/man/systemd.network.html>`

under Section Options.

Adding static routes

1. Connect to the SCG instance via SSH.
2. Edit the configuration file for the relevant network card and insert a section for each route according to the following example:

```
[Route]

Gateway=192.168.0.40

Destination=40.0.0.0/8
```

```
GatewayOnlink=yes
```

Note:

GatewayOnlink is required when you use DHCP and ensures that the route is set after the DHCP settings.

3. Restart the machine. Alternatively, restart the **systemd-networkd** service using the following command:

   `systemctl restart systemd-networkd`

To check the result, use the command ip route. Example:

`root@xxx-scg-1:~# ip route`

For the example above, it provides the following information : `40.0.0.0/8 via 192.168.0.40 dev internal proto static onlink`

Gateway configuration in the SCG WebAdmin interface

The configuration of your Scout Cloud Gateway includes several areas. The settings for authentication and connection of devices are required for successful on-boarding of new devices. Other settings, for example for forming clusters or accessing the instance itself, are optional.

- Log on to the SCG WebAdmin interface of your SCG instance and switch to the **Configuration** view.

The options are grouped in multiple sections.

Important

Your changes will not be saved until you click **Save**.

User authentication / Device certificates

For each device, after the onboarding process, the system creates a client authentication certificate so that users do not need to perform an additional logon for the SCG. For further information, see Device certificates.

Configure the validity of the device certificates and the behavior when the validity expires. Expiring certificates can be renewed automatically. Alternatively or additionally, you may manually renew certificates for selected devices in the **Devices** view.

1. Validity period for a device certificate in days.

   The minimum value must be 30 days and the maximum value must be 5 years (1825 days).
2. Device certificates are automatically renewed before their validity expires.(Once users re-connect with their device)

   This option is deselected by default.
3. Certificates that have already expired are also renewed automatically.

For further information, see Device certificates.

User authentication / Identity provider

Select and configure the identity manager you use for user authentication.

Active Directory

	OPTION	EXAMPLE	DESCRIPTION
(A)(The numbering (A)-(E) refers to the graphical representation of the connection process, see Integrating new devices)	Server	`ldaps://dc.int.sampletec-01.com:636`	LDAP protocol and FQDN, or the IP address of the AD server (Domain Controller)
			For LDAPS, you need a Base64-encoded CA certificate, see below.

	OPTION	EXAMPLE	DESCRIPTION
	Base	`DC=int,DC=sampletec-01,DC=com`	LDAP search base of the Active Directory domain
	Group template	`SCG_Group_`	Prefix used for the newly created AD groups
			The prefix is used to authenticate users and assign devices to OUs of the Scout Server.
			Only users who are members of an AD group with this configured prefix can log on to the Scout Cloud Gateway.
	CA certificates		Base64-encoded CA certificate (.pem for secure LDAPS communication (LDAP over SSL)
			To convert from .der to .pem, use openssl
			`openssl x509 -inform der -in cert.cer -out cert.pem`

- To check whether you can connect to the AD service with the specified AD data (without a new route), click **Test connection**. Enter the username and password of a test user and click **Connect**.

Note:

> The username must be specified as a user principal name (UPN). Example: mailto:mmi@int.sampetec-01.com

OpenID Connect / Keycloak

> **Note:**

> Keycloak support is deprecated. Please do not use it anymore.

> View the configuration data for your Keycloak client in the Keycloak administration console under **Installation > Format Option: Keycloak OIDC JSON**

For further information on configuring Keycloak, see Configuring Keycloak.

	OPTION	EXAMPLE	DESCRIPTION
	Keycloak realm	`ScoutEnterpriseCloudGateway`	Name of the realm you have created in the Keycloak administration console for your environment
	Keycloak server auth URL	`https://keycloak.int.sampletec-01.com:8443/auth`	Server authentication URL as specified in your Keycloak client on the **Installation** tab
			Note: For authentication via Keycloak, only HTTPS is supported. For this you need a CA certificate.
	Keycloak client ID	`SCG`	Name of your Keycloak client you have specified in

	OPTION	EXAMPLE	DESCRIPTION
			Keycloak under **Settings > Client ID**
	Keycloak client secret	`u14n936i-2642-43fc-o911-8n5de21a0e76`	Key generated by Keycloak for your Client ID, in Keycloak under **Credentials > Secret**

Scout Cloud Gateway authentication

The Scout Cloud Gateway supports a token-based logon for the registration of new devices. You may create as many tokens as you like and give them to users to register their device.

You can have a token created automatically or create it manually.

- To create a new token, click **Add token**. Assign an OU to the token.

1. Select Identity provider **Scout Cloud Gateway**.
2. Enter a new token (option A) A token must have a minimum of 16 and a maximum of 50 characters. Allowed characters:
   - alphanumeric characters `(a-z, A-Z, 0-9)`
   - special characters `(!?#%&()<=>@\^~*+,./:;_-)`
3. Generate a new token (option B)(from SCG 1 2209. In previous versions, for each new entry a token is generated automatically which you may replace with individual names (16 characters in length))
4. ID for the OU in which the user's device is to be registered during on-boarding
5. Delete token / entry
6. Create a new entry

User/password authentication

Up to eLux RP 6.10, the definition of SCG users with password was provided. For compatibility reasons, this method is still supported.

- Select **Support user/password authentication** and add new user entries.

Server device settings and VPN device settings

Server device settings

	OPTION	EXAMPLE	DESCRIPTION
(D)(1-The numbering (B)-(E) refers to the graphical representation of the connection process, see Integrating new devices.)	Scout Servers	`scout.int.sampletec-01.com`	FQDN or IP address of the Scout Server to which the devices need to be connected
			Multiple entries are allowed if there are several servers with access to the same Scout database.
	Gateway certificate		Certificate and key file for the communication between devices and Scout Cloud Gateway
			Only required if you use an individual CA certificate to encrypt the management protocol
			For further information, see Certificate-based management protocol in the **Installation** short guide.

	OPTION	EXAMPLE	DESCRIPTION
			The certificate must be issued for the gateway server.

VPN device settings

	OPTION	EXAMPLE	DESCRIPTION
(B)	Name servers (DNS)	`192.168.17.2`	IP address of the name server (DNS) that transmits the Scout Cloud Gateway to the eLux devices, enabling them to resolve the FQDN of the Scout Server name
			Multiple entries are allowed.
(C)	Routes	`192.168.17.0`	IP address range for which eLux will use the secure communication via the Scout Cloud Gateway
		`255.255.255.0`	Use this option for additional Scout Consoles
			Multiple entries are allowed.
(E)	VPN IP address range	`10.8.0.0`	Is configured in the **Gateways** view
		`255.255.255.0`	

SCG instance access

(only for VM installation)

For administrative access to your Scout Cloud Gateway instance via SSH, the public key of a generated key pair is required.

- To generate the key, use a suitable tool (such as PuTTY) and then copy the key content into the **Public key** field.

Multiple entries are allowed.

Gateways

Configure each new instance after installation in the **Configuration** view under **Gateways**.

VPN connections of a gateway instance

Each new SCG instance must be reachable for the target devices via VPN.

1. Select your new instance.
2. Variants for multiple gateway instances: Same host name and different ports or different host names and same port.
3. VPN port
    - Use a free port. The system does not check for existing assignments.
    - Open the port in your firewall.
4. Range of IP addresses that will be assigned by the Scout Cloud Gateway for secure communication with eLux® Example: `10.0.10.0`
5. Subnet mask for the IP address range Example: `255.255.255.0`

Web server configuration

During installation (from SCG 1 2209 also for the Debian installation) the system automatically creates a self-signed certificate that secures the connection between the web server and the of the SCG instance (from SCG 1 2101). This allows the administrator to perform further configuration using the WebAdmin interface - and an HTTPS connection - immediately after installing an SCG instance. The self-signed certificate is valid for 365 days and is renewed automatically.

If you have an individual certificate, replace the self-signed certificate. To do so, click **Select file** (only for VMs (.ova installation)).

For multiple gateway instances, you can either use a separate certificate for each instance and assign it in the respective WebAdmin interface. Alternatively, use a wildcard certificate for all instances, if this is applicable within your structure.

SCG administrators

The Scout Cloud Gateway comes with an **admin** account that has full access. Optionally, you can define additional administrator accounts. To each administrator, assign one of the following roles:

ROLE	RIGHTS
Admin	Unrestricted access
Token admin	Access is restricted to
	Can manage tokens
	the Devices view including certificate revocation
	the Info dialog with download link for the diagnostic files
	the User profile dialog
Operator	Access is restricted to
	the Devices view including certificate revocation
	the Info dialog with download link for the diagnostic files
	the User profile dialog

Defining new administrator account

1. Under **SCG administrators**, click **Add user**
2. Enter a username and a password for it. Repeat the password.
3. Select either the **Admin**, **Token Admin** or **Operator** role.

4. Confirm with **Save**.

The password is stored as a hash value in the database. The new administrator can log on immediately with his credentials.

STEP	OPERATION
1	Define a new account
2	Specify its username and password
3	Select a role
4	Delete this account

All accounts defined by you are listed with their username and role

In the **User profile** dialog the logged on administrator can see with which account and role he is logged on and since when.

Logging

1	Choose extent of diagnostic files
	In addition to SCG software components, the log level affects the OpenVPN logs
2	Configure thresholds for log rotation
	To download the diagnostic files, use the Info dialog.

Log level

- Choose the extent to which log and configuration files are retrieved and downloaded when you request the **Diagnostic files**.

LOG LEVEL	LOGS
Error	Only error messages
Info	Error messages and additional information (default)
Debug	Log files with extended content

The log files are stored under `/opt/unicon/scg/logs`

If you are running a cluster of multiple SCG instances, you only need to set the log level on one instance: the setting is automatically applied to the other instances. A restart for activation is not required.

Log rotation

Especially for log level Debug the amount of data increases quickly. Rotating the log files keeps the storage demand low. For each log file, the number of backup files and the size is limited. When the threshold values are reached, the files are rotated: A new file is created and the oldest backup file is deleted.

Log rotation is active by default and is automatically executed once per day.

The threshold values can be configured as follows:

OPTION	DEFAULT	VALUE RANGE
Maximum log file size in MB	10	10-100
Maximum number of backup files	5	5-10

The configured values are saved to the file logrotate.conf. This file and all rotated files in /opt/unicon/scg/logs are part of the **Diagnostic files**.

For further information, see Downloading diagnostic files.

Monitoring

The health check service allows administrators to check and monitor the availability and performance of an SCG instance and its internal components (backend, controller, MongoDB

controller, OpenVPN, ...) A health check request provides the overall status and the status of individual services as a result.

By default, the service is not active, but must first be started under **Monitoring > Health check**.

The health check is initially secured via an automatically generated token.

STEP	**OPERATION**
1	Activate health check
2	Optional: Define individual password
3	First save configuration, then click for a test request
4	Output: Overall status and status of each component

Upon request, the system generates status information in format on all relevant components.

- Status `0` means: No errors.
- If the overall status is `0`, all components are running.
- The time stamp specifies from when a component is up / down.

Calling the health check service

For a call, an HTTP GET is executed on the health check URL with the configured token given with it.

Example with curl:

```
curl 'https://<SCG URL>/healthcheck' -H 'Connection: keep-alive' -H
'Authorization: Bearer <token>' -H 'Accept: application/json, text/
plain' --insecure
```

The following parameters are used:

PARAMETER	DESCRIPTION
`<SCG URL>`	URL of your Scout Cloud Gateway
`<token>`	Token or password as defined in the SCG WebAdmin Interface under **Configuration** > **Monitoring**
`--insecure`	Needs to be specified if you use a Self-Signed certificate

For example, if Open VPN is not running, the result is as follows:


```
{
  "status": 1,
  "backend": {
    "status": 0,
    "since": "Tue 2021-10-19 12:21:24 UTC; 1 weeks 2 days ago",
    "message": "OK"
  },
  "controller": {
    "status": 0,
    "since": "Tue 2021-10-19 13:30:21 UTC; 1 weeks 2 days ago",
    "message": "OK"
  },
  "mongod": {
    "status": 0,
    "since": "Tue 2021-10-19 12:20:58 UTC; 1 weeks 2 days ago",
```

```
    "message": "OK"
  },
  "nginx": {
    "status": 0,
    "since": "Tue 2021-10-19 12:20:56 UTC; 1 weeks 2 days ago",
    "message": "OK"
  },
  "openvpn": {
    "status": 3,
    "since": "Wed 2021-10-20 07:08:52 UTC; 1 weeks 1 days ago",
    "message": "ERROR"
  }
}
```


Multiple gateway instances as a cluster

> **Note:**

> Clustering of instances is supported also for Debian installations from SCG 1 2209.

> By using clusters you achieve higher reliability and availability. If one Scout Cloud Gateway instance fails, another takes over. The second advantage is load balancing. Load balancing is controlled via the open VPN connections: When a device logs on, it is always forwarded to the gateway instance with the lowest load, meaning the lowest number of connections.

> **Note:**

> For reliable failover and fault tolerance, at least three instances are required in a cluster. This means that three databases (connected by clustering) are available on independent SCG instances in a so-called replica set. If you need more than three instances in a cluster, always use an uneven number.

> Per instance 1024 devices can be connected.

To use the Scout Cloud Gateway to connect devices from the Internet, there are several possible scenarios:

- Scout Cloud Gateway on one host as a single instance
- Multiple gateway instances as a cluster
- Operate multiple gateway instances in different networks separately
- Group multiple gateway instances into several clusters

Once you have multiple gateway instances on the same network, they can form a cluster. Communication between the instances is secured by a cluster key. In previous versions, this cluster key has to be added for a new instance to be part of the cluster.(see Defining a new cluster instance (earlier versions)) From SCG 1 2209 onwards, this is done in the background:

To add a newly installed gateway instance to an existing cluster (or instance), just use the WebAdmin interface of the existing cluster instance with the **Add gateway** function of the Gateways view (from SCG 1 2209). There you have to authenticate with the administrator password of the new instance. Once the new instance is visible, you may specify the public access data for VPN under **Configuration > Gateways**. For further information, see Adding a new gateway instance.

> **Important:**

> When you add a new instance in the **Gateways** view, its configuration is overwritten with the cluster configuration. Normally, this is an advantage because you do not need to configure newly installed instances when they are to be clustered.

> The configured public port must be opened in the firewall. All gateway addresses must be accessible from the outside.

The required network services are included.

> **Note:**
>
> When upgrading to a later SCG version, always upgrade all SCG instances of a cluster.

Adding a new gateway instance

Newly installed instances can be easily added in the WebAdmin interface of an existing (cluster) instance and thereby clustered with it.

> **Note:**
>
> The new instance must be installed ready and located in the same domain.

1. In the WebAdmin interface of the cluster instance, click **Add gateway**.
2. Specify the host name (without domain) or IP address of the new instance.
3. Leave the **Use initial logon data** option active, if you have not yet changed the password of the administrator account on the new instance.

   Otherwise, to authenticate, specify the credentials for the new instance.

   > **Note:**
   >
   > Any previous configuration data of the new instance are overwritten. As soon as the new instance is added to the cluster, the configuration of the cluster is active, including administrator accounts.

4. Confirm with **Yes**.
5. Switch to **Configuration > Gateways** and specify the public access data via VPN to the new instance. For further information, see Gateways.

The new instance inherits all the rest of the configuration of the cluster instance.

Client management and onboarding

New devices are configured to the gateway via the First Configuration Wizard.

The devices can be enabled to connect via intranet and via Internet,depending on where the user is located. For this, we recommend that you set up the network configuration in a way that the Scout Server name is resolved internally to the Scout Server and externally to the Scout Cloud Gateway. Users then only have to enter their logon data for the gateway the first time they use their devices outside the intranet. For further information, see General network settings.

Once successfully connected via the Scout Cloud Gateway, the devices remain connected to the gateway and retain their SCG configuration even in the following cases:

- A factory reset for the device is performed The option **Delete Scout Server address** must not be selected.
- Firmware update The option **Format system partition before update** may be selected.

> **Note:**
>
> If a device is connected via SCG, this information may be displayed in the Scout Console in the **Properties** window, as well as the SCG address (from Scout 15 2209 and eLux® RP 6 2209). These fields can also be used for creating records.

On-boarding new devices / standard

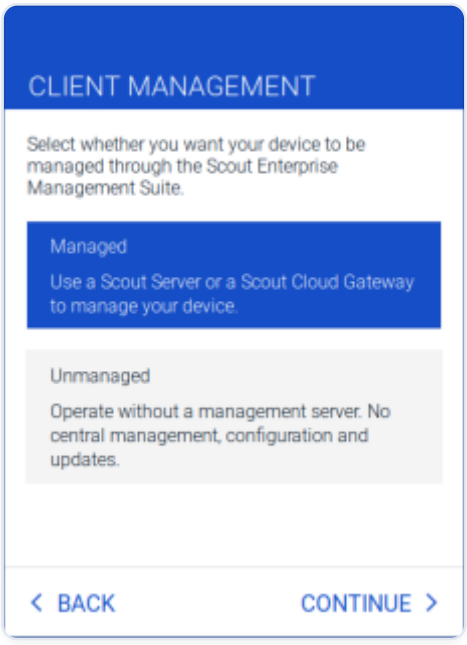
New devices that you want to connect via the Scout Cloud Gateway to your Scout Server need to be configured to the gateway (except when using uPilot) once via the First Configuration Wizard.

REQUIRES
The Scout Cloud Gateway is configured and users are authorized via one of the supported authentication methods.

1. Switch on your device. If the device is not in initial state, reset the device configuration. To do so, on the system bar, open the **Command panel** and click **Factory reset**.

   The First Configuration Wizard starts.
2. Select the display language and keyboard language.
3. Read and accept the license terms.

4. To configure the device's management via Scout Cloud Gateway, select **Managed**.



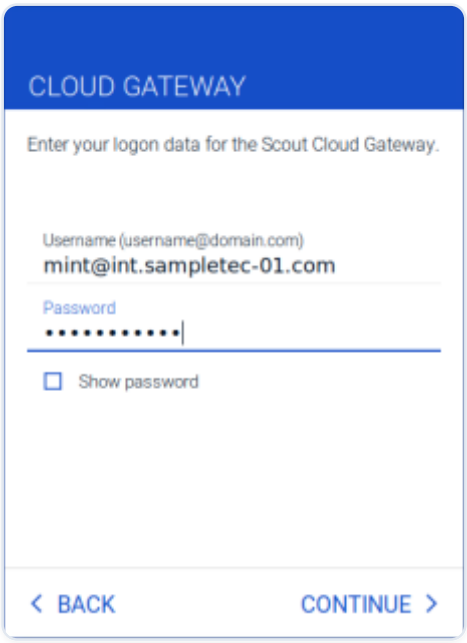
5. Enter the address of the Scout Cloud Gateway as FQDN or IP address.

> **Note:**
>
> If multiple gateway instances are operated in a cluster, it does not matter which of the gateway addresses is entered.

6. If Keycloak is used for authentication, you will be redirected to the Keycloak logon page. Specify and confirm your logon data.

   After authentication, you will return to the First Configuration Wizard.

7. If Active Directory is used for authentication, enter the AD user logon data. The username must be specified as a user principal name (UPN). Example: `mint@int.sampletec-01.com`



The relevant connection data are transferred via the Scout Cloud Gateway to the AD server. The AD user is checked, the Scout Server and OU are identified.

8. If the Scout Cloud Gateway authentication is used, enter the 16-digit token you have received from your administrator.

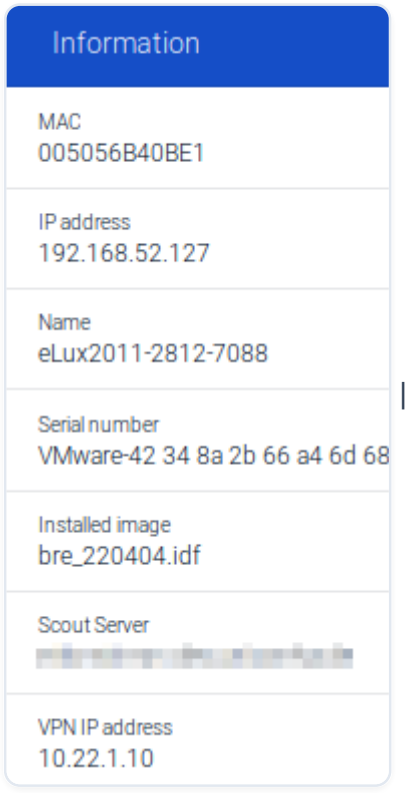
   The relevant connection data are transferred to the Scout Cloud Gateway and verified.

9. Click **Finish**.

*If the OU ID of the AD group or of the account definition for SCG authentication is invalid, the device enters the default OU that is set under **Preferences** > **Devices** in UniconOS Management Board. This is also true if authentication is done via Keycloak or token.*

*From UniconOS Management 2611, the device also enters the default OU if its OU is protected by a password or hidden, because the gateway doesn't send the password of the OU. To admit such devices as in earlier versions, set **Devices that join a protected or hidden OU without sending its password to Scout Server** to **Accept and log**, see Devices in the **UniconOS Management Board** guide.*

The gateway address, username, Scout Server, and assigned OU are displayed in a summary.



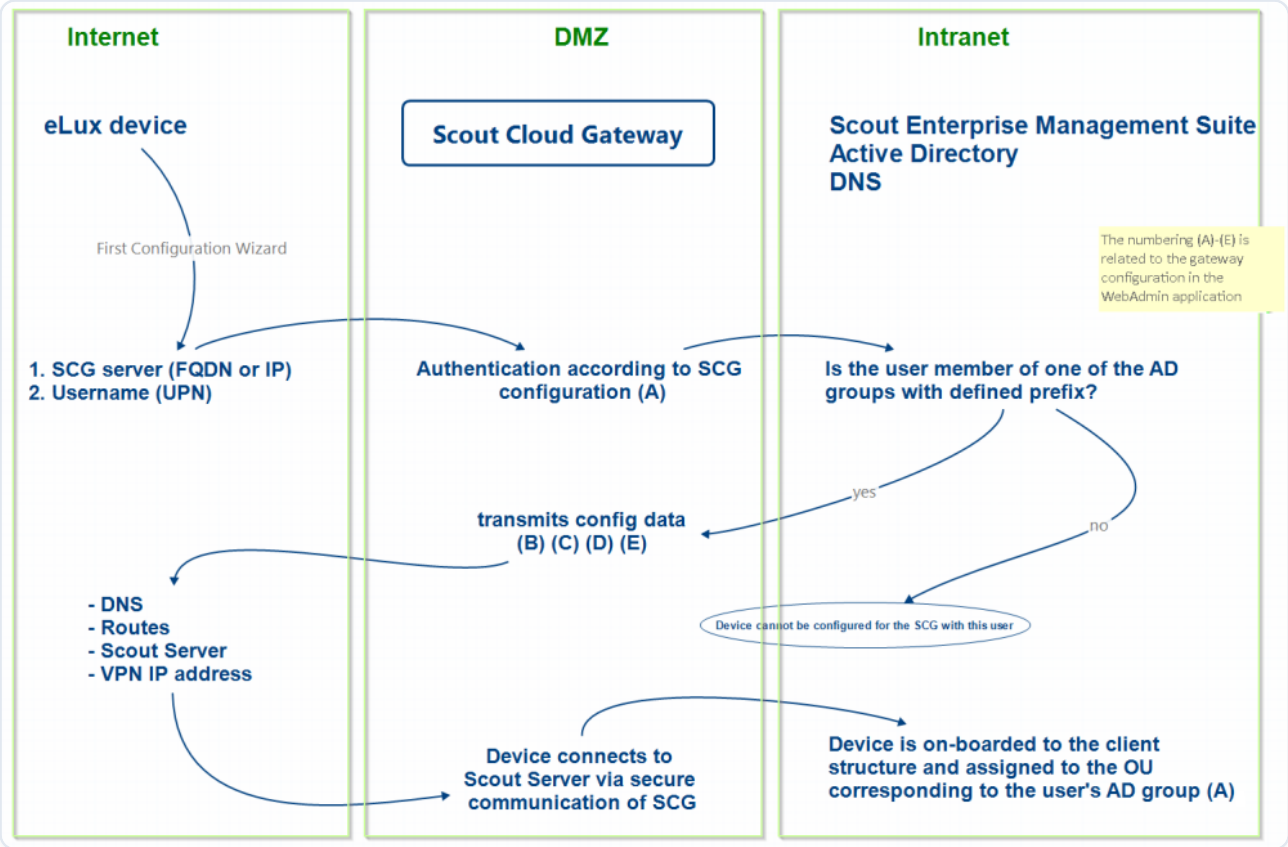
The configuration data are transferred from the Scout Cloud Gateway to the device and the device is restarted.

The device is connected to its Scout Server via the gateway. In addition to its local IP address, the device receives a VPN IP address.

The figure on the right shows the Configuration panel of the device after the First Configuration Wizard has been completed.

The device is shown in the WebAdmin interface in the Devices view with status connected (after short delay).

The flow chart below shows the communication of the onboarding process of a device via the Scout Cloud Gateway and via AD.



upilot Automated onboarding

(from eLux® RP 6 2302)

uPilot is a web service that simplifies onboarding of new eLux devices, for example for home devices, into a Scout infrastructure. The configuration of a new device is automated in a way that end users are able to on-board their device in a minimum of time, without having to know the address of their Scout Cloud Gateway. Especially when large numbers of devices need to be rolled out configured in short time, uPilot offers great support.

With uPilot, the assignment of the devices to their Scout Cloud Gateway is already defined before the actual roll-out:

uPilot is fed with a list of all devices that are to be connected. Customers specify the devices with a unique ID and a mapping to the URL of a Scout Cloud Gateway server.

As soon as a device with appropriate prerequisites starts the First Configuration Wizard, the following will happen:

The device sends its ID to the uPilot web service (1) which returns the URL of the associated Scout Cloud Gateway server (2). From the user's point of view, the only actions to be performed in the

First Configuration Wizard are to select the display and keyboard language and to enter the logon data (3).

Requirements

- Scout Cloud Gateway is configured and the user is authorized via one of the supported authentication methods.
- Software on the devices:
  - eLux RP 6 2302 or later In the eLux package **eLux Desktop Extensions**, the included feature package uPilot must be installed on the devices.

Customers provide the Unicon GmbH with a list of devices to be supported by uPilot and the associated Scout Cloud Gateway URLs. To do so, please contact the Unicon Support: myelux Support Case Form.

On-boarding a new device via uPilot

The following step-by-step instructions are from an end-user perspective.

1. Switch on your device. If the device is not in initial state, reset the device configuration. To do so, on the system bar, open the **Command panel** and click **Factory reset**.

   The First Configuration Wizard starts.
2. Select the display language and keyboard language.
3. Enter your user logon data. If Active Directory is used for authentication, the username must be specified as a user principal name (UPN). Example: `mint@int.sampletec-01.com`

   The relevant connection data are transferred via the Scout Cloud Gateway to the AD server. The AD user is checked, the Scout Server and OU are identified.
4. Click **Finish**.

The device is restarted and then connected to its Scout Server via the Scout Cloud Gateway. The assignment to the appropriate Scout Cloud Gateway was made in the background by uPilot.

SCG WebAdmin interface

In addition to configuration options, the WebAdmin interface of the Scout Cloud Gateway allows you to manage the connected devices and gateway instances.

The menu bar offers the following functions:

1. Switch between the views **Dashboard**, (from SCG 1 2209) **Gateways**, **Devices** and **Configuration**.
2. Info dialog with version information, documentation link and download link for diagnostic files.
3. **User profile** dialog with information on and functions for the logged-on user
    - Shows username, role and start of logon
    - Allows to switch the language (German/English)
    - Allows to **Change password**

For further information on the **Configuration** view, see Gateway configuration in the SCG WebAdmin interface.

Logon and language

For the initial logon to the WebAdmin interface the user account admin with an initial password is available, see Connecting and logging on to the SCG instance.

The admin account with global access rights can still be used with your own password. In addition, further users can be defined with the role **admin**, **token admin** or **operator**. For further information, see SCG administrators.

Failed logons

An account is temporarily locked for security reasons if too many incorrect password entries are made. By default, the account is locked for 10 minutes after three incorrect entries.

The number of incorrect entries and the lock time can be configured individually. To do so, edit the file `/opt/unicon/scg/config/config.json` and the following parameters:

```
[backend.lockUser.numFailingLogons]

[backend.lockUser.lockOutTimeInMinutes]
```

The new values are applied directly; a restart is not required.

Changing password

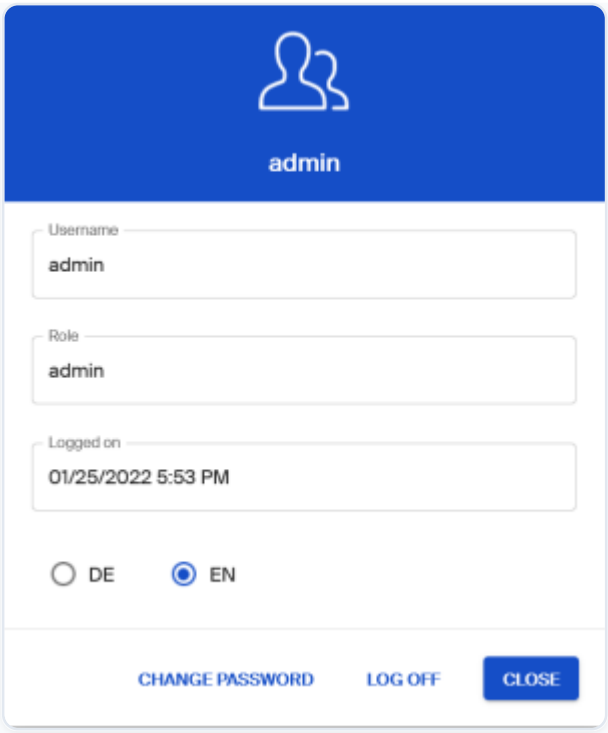
1. On the menu bar, click the **User profile** icon .
2. In the **User profile** dialog, click **Change password**.

Changing language

1. On the menu bar, click the **User profile** icon .
2. In the **User profile** dialog, choose between English and German.

The display language is immediately changed.

> **Note:**
>
> The language can be set individually for each user and is stored in the user profile.



General list functions (Devices and Gateways)

In the **Devices** and **Gateways** views, you can display the columns in any order and sort devices or gateways as you wish. You can also filter the display of devices, see Devices view.

All settings are saved in the browser used beyond the session. For this, local data storage must be allowed in the browser.

	DESCRIPTION
1	To sort by a field, click the column header. To change the sort order, click again.

	DESCRIPTION
2	To set the sorting order (or a filter) via menu click
3	To change a column width, drag the column header border to the desired position.
4	To change the order of columns, drag a column header to the new position.

Dashboard

In the SCG Dashboard, various widgets show the most important key figures and key performance indicators so that you can get an overview of the entire infrastructure at any time.

Devices

1	Total number of all connected devices
2	Currently connected and disconnected devices
3	Number of connected and disconnected devices
4	Devices that have not been connected for at least one month / at least one year
5	Devices whose device certificate has expired / will soon expire

Load Balancing

1	Total number of all connected devices with current distribution to instances

2	Number of gateway instances that are currently online, offline, or not configured
	Unconfigured gateway instances can be freshly installed instances or disabled instances.
3	Number of VPN servers on the gateway instances that are currently online, offline, or not configured
	If a VPN server is not configured, **Configuration > Gateways > Gateway-specific configuration** is missing.
4	Total number of possible device connections
	with display of instances that have only a few free IP addresses left

Performance

Average CPU and RAM usage, as well as memory across instances.

The instance with the highest values is displayed.

Configuration

1	Name of the current [Scout Server]. CA certificate or self-signed certificate that ensures secure communication between SCG and devices see **Configuration > Server device settings**
2	Selected Identity provider. For SCG authentication additionally number of token see **Configuration > User authentication**
3	Health check service active or inactive.Web server configuration see **Configuration > Monitoring** and **Configuration > Gateways**

4	Latest SCG version in the cluster
	Number of instances that are on earlier version
	Number of SCG administrators with roles see **Configuration > SCG administrators**

Administrator activities

Various activities are logged, including configuration value changes and adding/changing/deleting gateway instances.

The old values are crossed out, the new ones are underlined.

Gateways view

The configured gateway instances are shown with their properties. A newly installed SCG instance is configured under **Configuration > Gateways**. For further information, see Gateways.

NUMBER	DESCRIPTION
1	The role of a gateway instance, but also a state before a role is applied, are indicated by means of letters
	`P` Primary instance
	`S` Secondary instance
	`I` The instance is being initialized
	`E` The instance has an error or is not reachable
	`O` The instance synchronizes data (for example, when two instances are added at the same time)

NUMBER	DESCRIPTION
	The status of a gateway instance is shown as a colored icon
	▢ Up and running
	▢ Switched on, but deactivated: Only the existing connections remain active
	▢ Switched off / not ready for operation
2	Number of connected devices (from SCG 1 2209)
	will be updated once per minute
3	Public host name
	The VPN hostname and port are specified under Configuration > Gateways. (from SCG 1 2101)
4	IP address range for the VPN connections
	The VPN IP address and subnet mask are specified under Configuration > Gateways. (from SCG 1 2101)
5	Deactivate or delete a gateway instance (from SCG 1 2209 in one column)
6	Add a new gateway instance to the cluster (from SCG 1 2209)

Deactivating a gateway instance

Within a cluster, you can deactivate a gateway instance for maintenance purposes while existing device connections remain active. Only after the next restart the devices connect to another instance. New devices are also connected via another instance.

> **Note:**

> An instance is deactivated by removing its gateway-specific VPN configuration.

1. For the relevant gateway instance, on the very right under **Deactivate**, click **Deactivate gateway instance**.
2. Confirm with **Yes**.

The deactivated gateway instance is displayed without its VPN configuration.

- To re-activate the gateway instance after maintenance or after an update, under **Configuration > Gateways**, re-enter the public access information for VPN.

Deleting a gateway instance

With **Delete Gateway**, a deactivated instance is removed from the internal database. A deleted instance no longer is connected to the cluster.(For versions < SCG 1 2209, the cluster information is retained and when the physical instance is restarted at a later time, a new clustering is triggered in the network)

REQUIRES
The cluster has more than one instance.

1. Make sure that the gateway instance you want to delete is deactivated.
2. For the relevant gateway instance, under **Delete gateway**, click **Delete**.
3. Confirm with **OK**.

> **Note:**

> After confirmation, the system even removes the instance you are currently logged on to.

> You can continue to use a deleted instance separate from the cluster. Note that re-clustering a standalone instance (via the **Add Gateway** button) would also require re-onboarding the devices connected to the instance.

Adding a gateway instance

You easily add a newly installed instance to an existing instance or cluster so that the new instance inherits the entire configuration of the cluster. For further information, see Adding a new gateway instance.

Devices view

All integrated devices are shown with their properties. The **Devices** view gives you an overview of all devices in your infrastructure. You can filter the devices on one or more properties. For all or selected devices, you can perform actions and, for example, renew certificates.

For further information, see Device certificates.

NUMBER	DESCRIPTION
1	Number of all devices integrated in this SCG instance, number of connected devices, number of disconnected devices
	If a filter is active, the numbers refer to the filtered values.
2	Select all devices
3	Select individual device
4	Status of a device: connected or disconnected
5	To sort by a property (ascendent, descendent, undo sorting), click the relevant column header
6	Point to a column header to display filtering and sorting functions.
7	Remove all filters
	As soon as a filter is active, the icon is displayed in color.
8	Renew or revoke certificates for selected devices

NUMBER	DESCRIPTION
9	Remove selected devices

Virtual IP address

The virtual IP address is the internal IP address assigned by the VPN for the duration of a connection. It is displayed once a device is connected and the VPN tunnel is established.

> **Note:**

> For clusters: When you upgrade to a later SCG version, always upgrade all SCG instances in a cluster. Otherwise, the virtual IP address is displayed only for devices that are connected to an SCG instance with version SCG 1 2201 or later.

> The virtual IP addresses also help you keep track of the boundaries of your VPN address ranges.

Filtering devices

Filter rules allow you to filter the devices on any fields (text and date). Multiple filter rules can be combined.

- A filter rule refers to one field and is displayed in one line.
- Additional filter rules referring to other fields may be added and must be linked by an operator (`And` / `Or`).
- The filter rules are dynamically applied to the data.

	DESCRIPTION
1	Field by which is filtered
2	Operator for date or text fields
3	Filter value
4	For date fields, either select or enter the date.
5	Operator for linking another filter rule (`And` / `Or`)

	DESCRIPTION
6	Delete individual filter rule
7	Add more filter rules

1. On a column header, click the ⋮ and then select **Filter**.

   A filter definition is displayed in the left section of the table, which is still incomplete. The selected column is already displayed.

2. Check that the desired field is selected under **Column** (1). If necessary, select the field you want to filter by from the drop-down list.
3. Select the **Operator** (2) using the drop-down list. Different operators are available for date fields (example: is before) than for text fields (example: contains).
4. Specify the **Value** (3) you want to filter on.
5. To filter by another field, click **Add Filter** (6) and specify whether the two filter rules should be linked by `And` or `Or` (5).

*The data is immediately and dynamically filtered by the defined filter rules. The **Remove filter** button becomes active.*

> **Note:**
>
> Defined filters remain persistent for longer than the current session. You can tell whether a filter is active by the blue color of the button .

- To delete all filter rules, click the **Remove filter** button on the right above the table.

Removing devices from the SCG integration

The devices may be connected or disconnected.

1. In the selection column on the far left, select one or more devices.
2. In the upper right corner, click **Remove devices**.

The devices are no longer connected via the SCG, only the currently existing connections remain unaffected. Certificates that are still valid are automatically revoked. To reconnect, the devices must go through an onboarding process.

Device certificates

For each device, after the onboarding process, the system creates a client authentication certificate so that users do not need to perform an additional logon for the SCG once they have registered their device. As long as the certificate is valid, users devices automatically connect to their Scout infrastructure via the SCG. Users only log on to Scout with their usual account, for example an AD account.

These certificates are called **Device certificates**. You can preconfigure their validity and expiration behavior for your SCG instance. For further information, see User authentication / Device certificates.

Validity period

The validity period of the device certificates is set to three years by default. It must be 30 days minimum and 5 years (1825 days) maximum. Within this range, you can freely define the validity period. The expiration date (**Validity**) of each device certificate is derived from the defined validity period. The validity date is set in the following situations:

- When a device is on-boarded
- On automatic renewal of a device certificate (if configured)
- On manual renewal of a device

You can view the validity date of a device certificate at any time in the certificate details under **Properties**. To do so, click the certificate icon. If a certificate is registered for renewal, you will see the new validity date under **Renewal**.

> **Note:**
>
> Defining a limited validity period without automatic renewal allows you, for example, to make the devices reboot after a certain time.

Automatic renewal

The device certificates can be renewed automatically before they expire. This function is not active by default. You can configure it in two variants: With the option **Automatically renew before expiration**, the system attempts to renew the device certificate seven days before expiration. If the user does not connect with their device within these seven days, the certificate expires. Only if the option **Allow renewal of expired certificates** is also active, the expired certificate will be renewed automatically on the next device start and contact to the SCG.

Note:

Automatic renewal (including expired certificates) allows you to avoid user interaction and the need for on-boarding a second time.

With automatic renewal configured, the following situations result for a device within the four stages shown:

TIME	---------->	---------->	---------->	---------->
Stage	Onboarding	30 days before exp.	7 days before exp.	Expiration
Icon	▢	▢	▢	▢
Authentication for config 1	✓	✓	✓ Certificate is renewed	X
Authentication for config 2	✓	✓	✓ Certificate is renewed	✓ Certificate is renewed

Config 1: Option **Automatically renew before expiration** is active. Config 2: Option **Allow renewal of expired certificates** is additionally active.

Note:

After you revoke certain certificates, they will be excluded from automatic renewal.

Manual renewal

By default, device certificates expire automatically. Of course, you can also avoid re-onboarding known devices by renewing the device certificates manually in the **Devices** view. Again, the

certificate icons are displayed in color and allow you to see at a glance whether action is required. Note the following:

- When you set a manual renewal, the validity of the relevant certificates is extended by the validity period defined in **Configuration > User authentification**. The new date is displayed in the certificate details under **Renewal**. The **Devices** view still shows the original date, but the color of the certificate icon changes to blue.
- After you set a renewal, the certificates of the relevant devices will be renewed on their next contact to the SCG. After that, the new validity date is displayed in the **Devices** view.
- A device that users log on with after the manually defined expiration date (for example because the mobile/home device has not been used for a long time) is denied. Either the device needs to be on-boarded again or the admin needs to renew the certificate manually.

Manage device certificates

In the **Devices** view, administrators may view, revoke or renew the certificates for their devices at any time. The certificates are displayed differently depending on their validity:

Certificate is valid more than 30 days

Certificate will expire in less than 30 days

Certificate has expired

Certificate is not available or has been revoked

The same colors are used for the button to revoke a certificate.

1	Expiration date for the validity of a device certificate
2	View details of a device certificate
3	Revoke certificate of a device
4	Renew or revoke certificates of multiple devices
	The devices must be selected beforehand.

- To select either individual or all devices, use the selection column on the left side of the table.

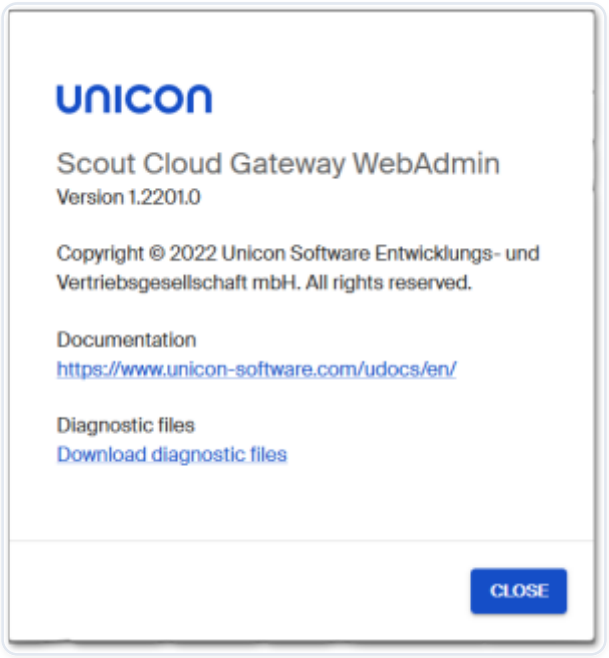
For further information, see Devices view.

Downloading diagnostic files

In the WebAdmin interface of the Scout Cloud Gateway, you can download all relevant diagnostic files and have a look into the communication between the Scout Cloud Gateway and the device. The diagnostic files provide support staff with helpful information and help identify missing routes or not properly working services.

The scope of the diagnostic files (**log level**) can be defined in the gateway configuration under **Logging**. For further information, see Logging.

1. On the right of the menu bar, click ⓘ.
2. In the **Information** dialog, click the **Download diagnostic files** link.



A set of log files is compiled and downloaded in a .zip file.

3. Download the .zip file.

Access diagnostic files without WebAdmin interface

If the WebAdmin interface is not available, you need to access the instance directly to run the diagnostic script. Provided the public key for SSH access is installed, you can access the machine with `ssh <IP or host name>`. Then, in the relevant directory, run the script that creates the diagnostic files and retrieve the created .zip file:

```
cd /opt/unicon/scg/scripts

./createDiags.sh



scp root@<IP or hostname>:/opt/unicon/scg/scripts/scg-
diags_<timestamp>.zip
```

If the public key for SSH access is not installed, connect to your machine locally or through vCenter.

For the Debian installation, the script is located in the same directory. However, it creates a smaller .zip file, as system files are not retrieved.

ELIAS 18

ELIAS 18 is the platform-independent, web-based front end of the eLux® Image Administration Service (ELIAS).

Note:

ELIAS 18 isn't part of the standard installation of UniconOS Management. It's discontinued with UniconOS Management 2611 LTSR and isn't developed further. ELIAS 18 can't build images that contain UniconOS 2612 or later. From UniconOS Management 2611 LTSR, image management is integrated in UniconOS Management Board, see Software in the UniconOS Management Board guide. To take a container over, export it in ELIAS 18, and import it in **Software**, see Importing a container. For details, see Deprecation.

Managing images and templates

To work with an image or template, in the overview of the **Image Library**, click to open it. The operations described below can only be applied to open (and saved) images or templates.

1	Back to the overview of the **Image Library**

2	Rename image or template
	Click the name and enter a new name.
3	Image or template is unlocked
4	Image or template is complete
5	Preferences
6	Lock/unlock image or template
	To locked images or templates no changes can be made.
7	Sign image or template
	Requires a code signing certificate. For further information, see Signing an image.
8	Export an image with or without packages
	The **Export** button is only shown if the image or template is saved. As soon as you make a change, the **Save** button is shown instead.
	For further information, see Exporting an image.
9	Delete the image or template from the container
10	Save a copy of the image or template under a new name
11	Solve the image or template
	For further information, see Creating an image.

Locking an image

Once you have created a template which is perfectly suited as a basis for creating images, you have the option to lock it. This means that your template will be write-protected and no further changes can be made. The same applies to images: Once you have created an image ready for roll-out, you can lock it. No more changes can be made.

REQUIRES
Admin or **User** role

1. In the **Image Library**, open the relevant image or template. Make sure it is saved.
2. Click **Lock**.

*No more changes can be made. The only remaining actions are **Unlock**, **Export** and **Save as**. Note that the **Lock** icon next to the name has changed.*

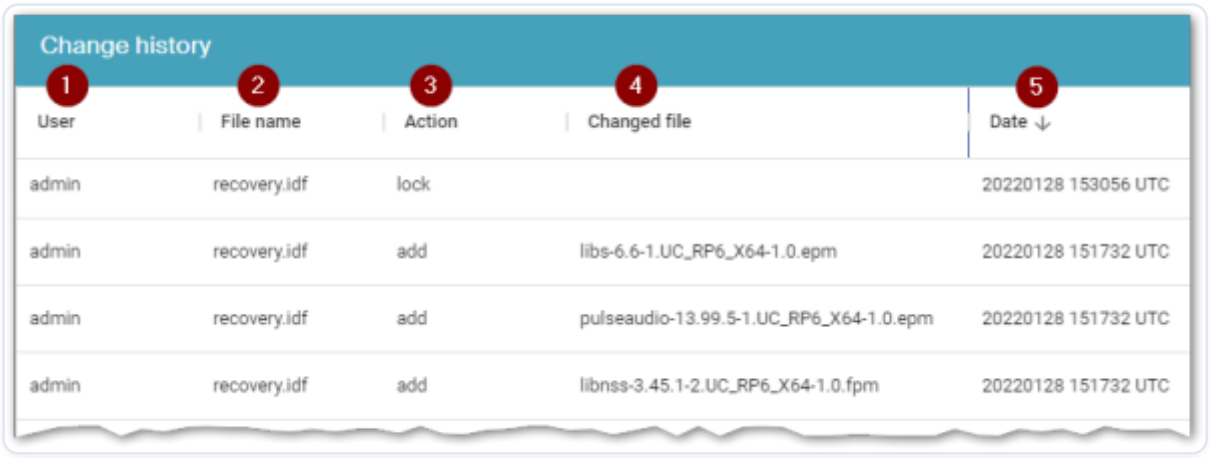
Tracking and viewing changes

All changes made to images and templates are tracked and reported in a change history.

REQUIRES
Admin or **User** role

1. On the ELIAS menu bar, click .
2. Filter and sort the columns as required (see screenshot).
3. To download the change history, click **Export**.

The change history lists all actions executed on images and templates (3) with their user (1), relevant image or template (2), changed file (4) and time stamp (5).



ELIAS

The eLux® Image Administration Service (ELIAS) allows you to manage the firmware of devices running eLux.

> **Note:**
>
> This guide describes the legacy ELIAS, which is end of life. The legacy ELIAS is no longer part of the standard installation from UniconOS Management 2611 LTSR, and an upgrade to 2611 removes it. UniconOS Management 2609 is the last release that installs it. The legacy ELIAS can't build images that contain UniconOS 2612 or later. To build your firmware images, use the image management that is integrated in UniconOS Management Board, see Software in the UniconOS Management Board guide. For details, see Deprecation.

IDF

An image definition file is a structured package list defining all software packages for the operating system and applications.

One of the basic ELIAS features is the built-in mechanism checking for conflicts and dependencies regarding other packages. Required packages are automatically identified and requested. They can easily be added with just one click. This prevents software incompatibilities in the image definition file.

Creating a new IDF

Updating existing IDFs

Saving an IDF

Signing an IDF

Exporting an IDF

Printing an IDF

Enabling Chrome Enterprise Premium (CEP) on eLux®

Chrome Enterprise Premium (CEP) hands Google Chrome extension control over to your Google Admin console. By default, eLux blocks all Chrome extensions on the device, including the ones CEP needs, such as Secure Enterprise Browser and Endpoint Verification. Enabling CEP removes this local block.

You enable CEP with a single option in the image definition file (IDF). There is no separate setting in Scout or on the device itself.

> **Note:**
>
> CEP is available from Google Chrome version 149.0.7827.196.7.2607.0-3 onwards. It is switched off by default.

1. In ELIAS, open the IDF used by the devices you want to change.
2. Expand **Google Chrome Web browser**.
3. Tick **Chrome Enterprise Premium Support**.
4. Tick **Native Messaging Host for Google Chrome**.

   CEP requires this feature, and it also keeps single sign-on working.
5. Untick **SSO Support for Google Chrome**.

   CEP and SSO Support cannot be used together. CEP already covers single sign-on for Citrix Secure Access™.
6. Save the IDF.
7. In Scout, assign the updated image to the device or OU and start the update.

   The device restarts as part of the update.
8. In your Google Admin console, force-install the extensions you want, for example Secure Enterprise Browser and Endpoint Verification, for the relevant users or devices.

> **Important:**
>
> Enabling CEP on eLux only removes the local extension block. The extensions themselves are still delivered by your Google Admin console. Skipping this step means no extensions are installed.

For further information on editing an image definition, see Updating existing IDFs.

For what CEP changes on the device and how to force-install extensions, see Chrome Enterprise Premium (CEP).

Scout Report Generator

The Scout Report Generator is used to create reports based on devices, OUs, applications or asset entries currently stored in the Scout database.

> **Note:**
>
> The Scout Report Generator is end of life. It is no longer part of the standard installation from UniconOS Management 2611 LTSR, and it is not available for UniconOS Management 2611. UniconOS Management 2609 is the last release that installs it. Use UniconOS Management Board instead: you can filter its device lists and export them as CSV files, see Display devices. For details, see Deprecation.

Report definition

Defining a report includes two basic steps:

- The layout of a report specifies the relevant database fields and their display properties. In addition, it determines the way the report output is formatted, including the font, page layout and headers.
- The filter is used to filter the selected fields for particular values. You can combine several filter expressions by using operators.

Dynamic Device Groups

Dynamic Device Groups enable administrators to run cross-OU commands for freely definable device groups. For example, you can send a message to all devices with a specific image throughout the whole organization. Or you can run a update on all devices with a specific version, across all OUs. Even to another can be applied to a Dynamic Client Group. The following features can be applied:

- Commands
- Configuration run
- Notifications for software deliveries or firmware updates
- Notifications for relocation

Dynamic Device Groups are based on reports created in the . The reports are exported to the once, and from that point onward, are displayed as a . Commands applicable to OUs or to individual devices can also be applied to .

The report layout must include the field. The report type must be a list of **Devices** or **Assets**. For further information on defining , see Creating Dynamic Device Groups in the Scout Report Generator guide.

Dynamic Device Groups are displayed in the Scout Console in a special window and remain there for re-use until they are deleted. They can be updated any-time with a click.

User rights

When you create Dynamic Device Groups, the user access rights are respected as defined in the administrator management.

By default, executing commands and other functions on a Dynamic Device Group is subject to the object rights configured for the relevant devices and OUs. This can result in a function not being executed because at least one device does not have the required object rights. To avoid this, disable the check for the underlying object rights under **Security** > **Manage administrator**s > **DCG rights**.

Creating Dynamic Device Groups

Creating Dynamic Device Groups consists of two steps that are carried out in Scout Report Generator:

- Creating a report filtered by the desired properties
- Exporting the report as a Dynamic Client Group to the Scout Console

CREATING A REPORT AS BASIS FOR A DYNAMIC CLIENT GROUP

1. Select **File** > **New report** or click the **New report**.
2. In the **New report** dialog, select the **Report type** Create list of devices or Create list of asset entries and confirm with **OK**.
3. For the selected report, on the toolbar, click the **Layout** button to define the report layout.

   In the **Report layout** dialog, on the **Columns** tab, choose those database fields you want to filter by.

   > **Important:**
   >
   > Make sure to include the Client Identifier field. This field is required for unambiguous identification of the devices.
   >
   > Specify the desired layout properties.

   For further information, see Report layout.
4. For the selected report, on the toolbar, click the **Filter** button and define the filter criteria.

   Example: OS version LIKE %4.8% extracts all devices with an OS version 4.8 installed.

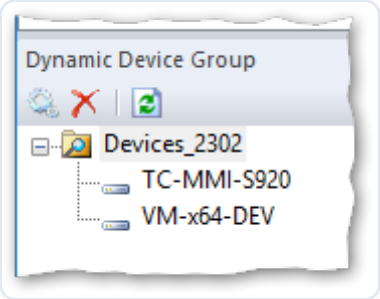
   For further information, see Report filter.
5. Run the report by clicking the **Create report**.

   The report output is created and displayed on the right.
6. Save the report.

EXPORTING A REPORT AS DYNAMIC CLIENT GROUP

1. Select the report that you want to export.
2. Click **Export** > **Dynamic Client Group** to start data export to the Scout Console.

The report is exported and provided as Dynamic Client Group in the Scout Console in the **Dynamic Device Groups** window. All available commands can be applied to the Dynamic Client Group.



For further information, see Using Dynamic Device Groups in the **Scout** guide.

Scout Agent for Windows

Scout Agent for Windows allows you to manage devices running Windows operating systems, such as Windows 10 IoT Enterprise, with the Scout Enterprise Management Suite®.

Windows platforms

Scout Agent for Windows platforms allows you to manage devices running Windows operating systems such as Windows 10 IoT Enterprise by using Scout Enterprise Management Suite. Scout Agent for Windows is charged with the communication between the Windows devices and Scout Server.

In addition to asset management, device configuration and help-desk features you can perform system updates on the devices by using Scout. For version 1.3 and later versions, Scout Agent for Windows can also provide applications defined in Scout on Windows devices.

Licensing

In addition to the Microsoft Windows license for the operating system the client requires a Scout Management license for device management by Scout.

To simplify license distribution, if your devices are not delivered with integral licenses, you can use so-called **loose licenses**. These unallocated licenses can be distributed by the Scout Server to unlicensed devices.

> **Note:**

> Without a management license, a client cannot be run in **Managed** Mode.

> For further information, see Scout Management license in the **Licensing and Subscription** White Paper.

Installation and interface

- Installing Scout Agent for Windows
- Configuring Scout Agent for Windows
- Configuration window

Installing Scout Agent for Windows

1. Download the latest Scout Agent for Windows version from our technical myelux.com portal and unpack the ZIP file.
2. Run the `ScoutAgentBundle.exe` file and follow the instructions of the installation wizard.

After completed installation the Windows taskbar shows the systray icon Scout Agent.

Unattended installation

For unattended installation, use the command-line parameters:

```
install /quiet
```

Example:

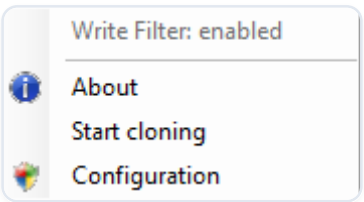
```
ScoutAgentBundle /install /quiet
```

Configuring Scout Agent for Windows

> **Note:**

> The Scout Agent for Windows supports the languages English and German.[1[The shown language is determined by the display language selected in the Windows control panel.]]

1. Show all systray icons of the Windows taskbar and right-click **Scout Agent > Configuration**.



2. Click **Configuration**.

The Scout Agent for Windows is loaded and requests the device password.

3. Enter the device password of the client.

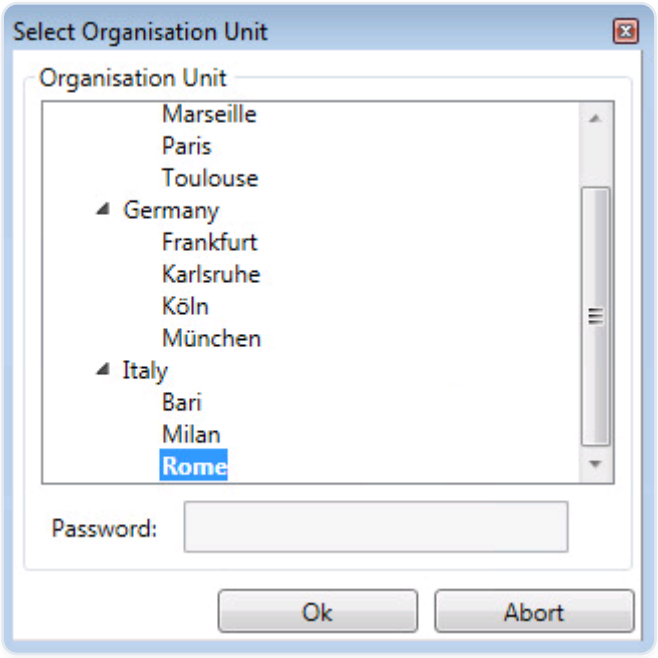
   *For initial start-up the password is elux. During first connection to the Scout Server, the password is changed to the value defined there (see **Scout Console > Options > Base configuration > Security > Local security settings > Device password**).*

   *The **Scout Agent for Windows Configuration** dialog opens. The Scout Server address is predefined as ScoutSrv. For further information, see Self-registration of devices in the **Scout** guide.*

4. In the **Scout Server Address** field, enter the IP address or FQDN of the Scout Server.

   The Scout Agent for Windows sets up a connection to the Scout Server.

5. Next to **Organisation unit**, click the ... button to display the **Select Organization Unit**dialog, and then assign the device to an OU.



6. Confirm with **OK** and **Apply**.

The device receives the configuration data of the selected OU from the Scout Server. The configuration becomes active after the next restart.

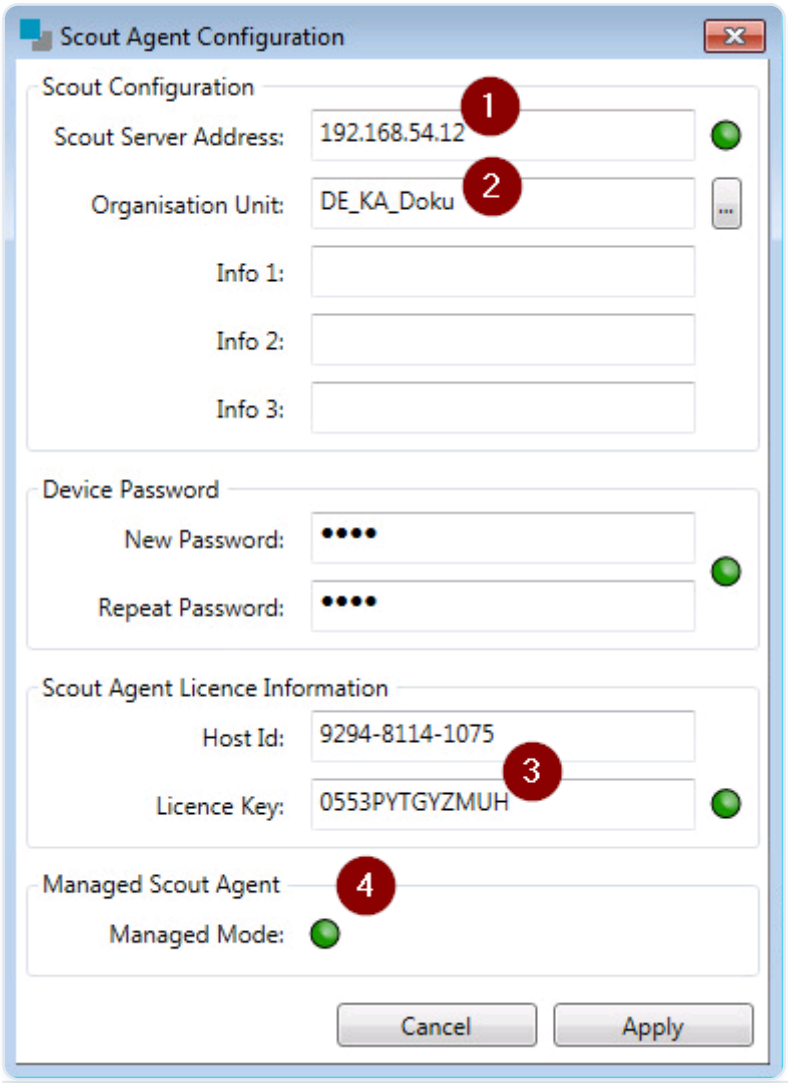
Note:

On each client restart, the Scout Agent for Windows is started automatically. As long as the Scout Agent for Windows is active, the **Scout Agent** is displayed in green.

Configuration window

After you have installed the Scout Agent for Windows, the Windows taskbar shows the systray icon **Scout Agent**.

To open the configuration window, click the systray icon.



	LEGEND TO NUMBERS
1	IP address or FQDN of the connected Scout Server
2	OU the Windows client is assigned to
3	License key, only required for Embedded 7 clients with pre-installed image

	LEGEND TO NUMBERS
4	If the Windows client is connected to the specified Scout Server, and if it is assigned a Scout Management license, the **Managed** mode is active.

Scout management features

By using Scout Enterprise Management Suite® you can manage different aspects of Windows devices.

REQUIRES
• The devices are provided with the required licenses.
• The devices are connected to a [Scout Server] and are in **Managed** mode.

Scout Agent for Windows controls the communication between the Windows client and Scout. A systray icon helps you access the Scout Agent for Windows configuration at any time. The important point here is that the **Managed** mode is active.

If you have used the client without Scout so far, you only need to specify your Scout Server and, if required, enter a license key in the **Scout Agent Configuration** to enable **Managed mode**.

If, on the Windows system, relevant software components or required access rights are missing, running Scout management features may be restricted.

Security

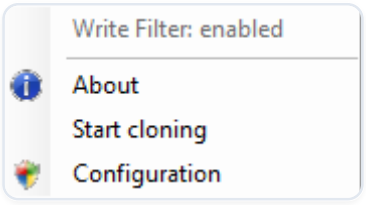
- File-based write filter
- Device password
- Mirroring

File-based write filter

The file-based write filter (FBWF) is a device configuration option that you can enable and disable in the Scout Console (**Setup > Hardware**).

Changing the status of the write filter requires a restart of the client. So do any modifications on the configuration. For further information, see Device configuration.

The current status is shown in the systray context menu of the .



> **Note:**

> The write filter status is determined by the Scout Agent for Windows. If the client is not in **Managed** mode, the Scout Agent for Windows service is not started on the client. In this case, the context menu of the Scout Agent for Windows displays the status `Write Filter: Unknown`.

> Alternatively, you can retrieve the write filter status from the command line running the following command as administrator: `fbwfmgr /displayconfig`

Device password

You can change the locally saved device password in the Scout Console for all devices:

- Click **Options > Base device configuration > Security > Local security settings > Device password**

> **Important:**

> By using this feature you change the device password of **all** devices managed by this Scout Server.

> The password is required to access the **Scout Agent for Windows configuration** dialog (systray icon **Scout Agent > Configuration**).

Moreover, the device password ensures that the client can only be managed by a Scout Server provided with the same password.

For further information, see Local device password in the **Scout** guide.

Mirroring

When you establish a mirror session, all security options apply that have been set in the Scout Console for that OU (for Scout Agent for Windows 1.3 and later versions).

During the mirror session a system message is shown on the client device allowing the user

- to stop mirroring at any time
- to grant only read access to the administrator (for Scout Agent for Windows 1.3 and later versions).

For further information, see Mirroring client screen.

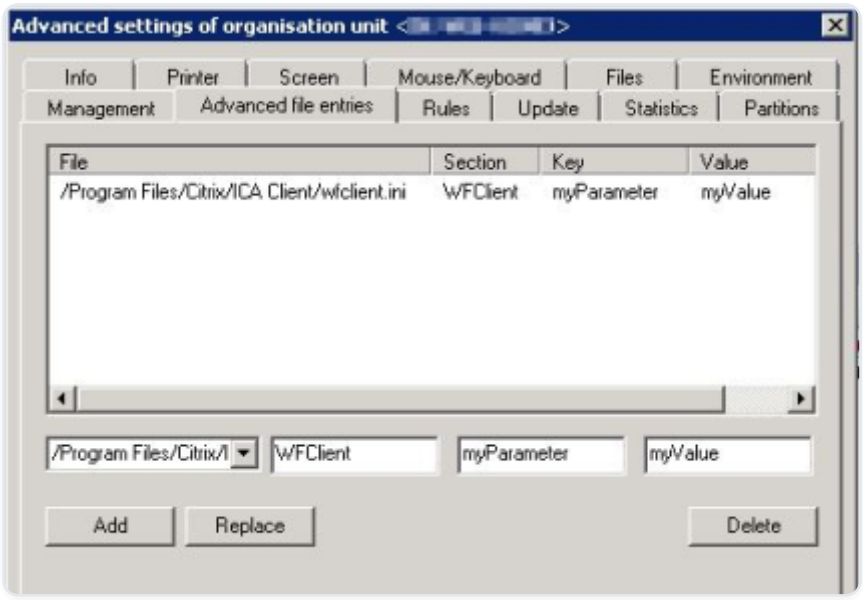
System updating

By using UniconOS Management Board, you can provide your Windows devices with firmware updates. Also, you can modify configuration files and settings of the Windows client, and transfer any files you want to have on the client.

- Advanced file entries
- Transferring files to the Windows client
- Firmware update

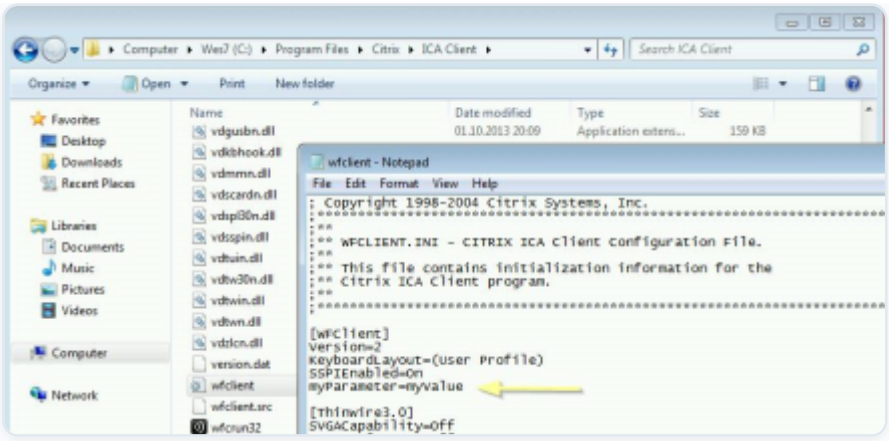
Advanced file entries

By using the **Advanced file entries** feature of the Scout Console, you can set and modify parameters and their values in `.ini` files. This feature can also be used to modify particular preferences and parametrize program calls on the Windows client.



Note:

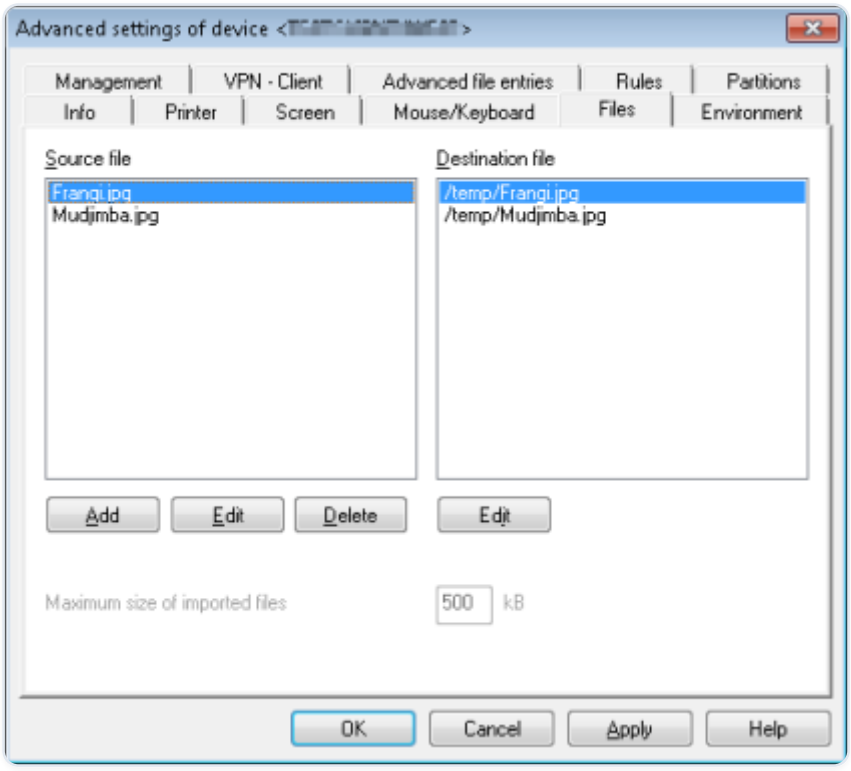
For the file path on the Windows client, do not enter a drive letter (it is always drive C:) and use slashes instead of back slashes.



For further information, see Advanced file entries in the **Scout** guide.

Transferring files to the Windows client

By using the Scout feature **Files configured for transfer** in the console, you can transfer any files from Scout to the Windows devices.



Note:

For the file path on the Windows client, do not enter a drive letter (it is always drive C:) and use slashes instead of back slashes.

For further information, see Files configured for transfer in the **Scout** guide.

Firmware update

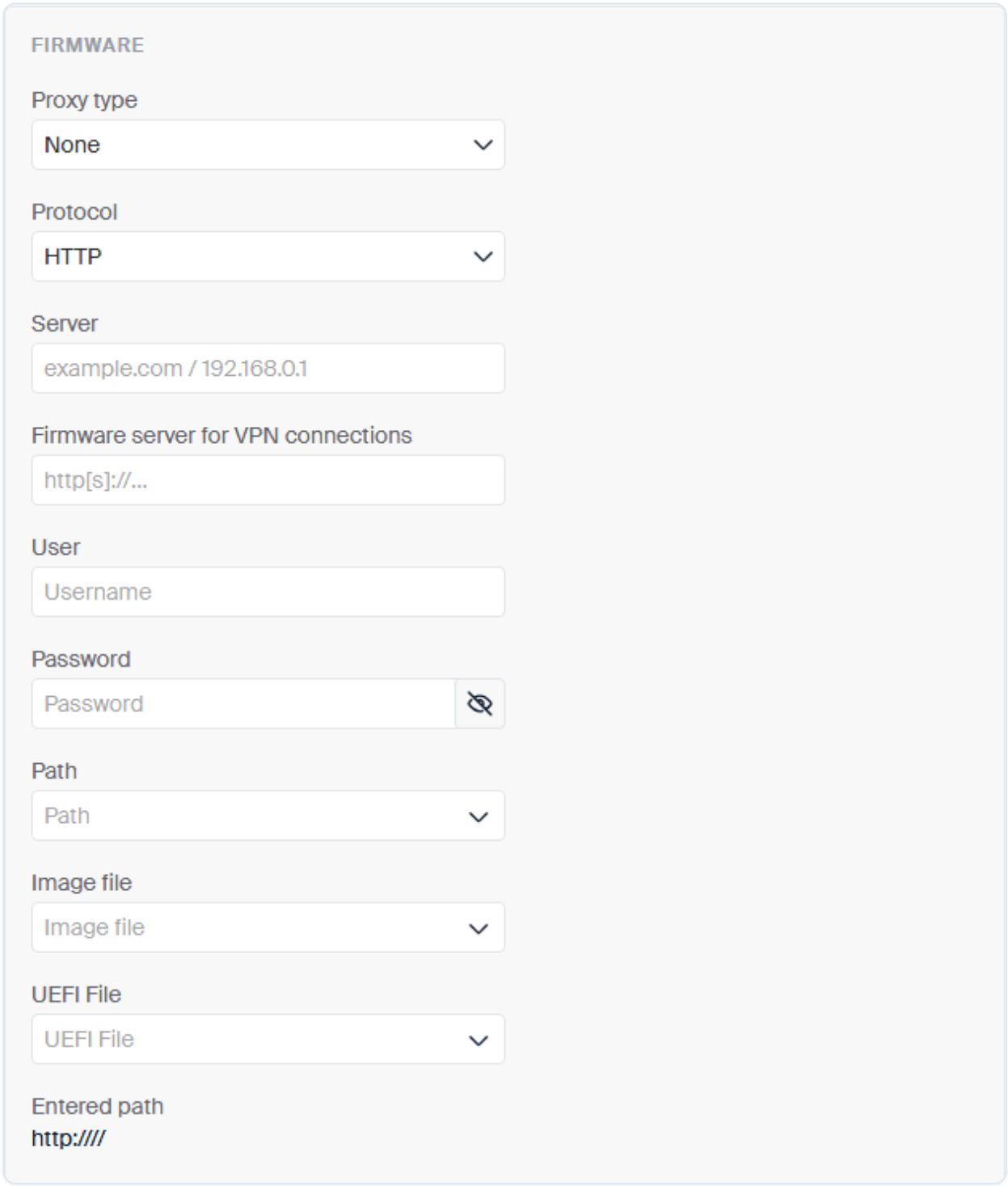
Installing further software on your Windows devices is done by performing a firmware update. For that purpose, a ZIP file is transferred containing an installation method and the software to be installed. The firmware update is configured in UniconOS Management Board, in the **Firmware** section of the device configuration of the relevant devices.

> **Note:**
>
> Due to security reasons, we recommend that you first thoroughly test the installation method in combination with the software to be installed on a reference system.

CONFIGURING FIRMWARE UPDATE

1. In UniconOS Management Board, open the device configuration of the relevant devices, and click **Edit**. In the list of sections on the left, select **Firmware**.
2. In the **Image file** field, specify a ZIP file that contains an installation method in the form of the file `install.cmd` and the software to be installed.

   If the complete path name is required, you must use the environment variable %CD% in the installation method `install.cmd`. Example for installing a Microsoft hotfix with Windows Update Installer: `wusa.exe %CD%windows6.1-KB28xxxxx-v2-x86.msu /quiet`
3. If you want to allow the users to postpone updates, set **Number of allowed deferments** and **Delays until next reminder**, see Reminder.
4. Click **Apply** (base configuration or an OU) or **Apply settings** (a device).



For further information, see Firmware server.

5. For the relevant devices, perform an **Update software** command. For further information, see Remote commands.

UNINSTALLING SOFTWARE

1. For the relevant devices, in the **Firmware** section of the device configuration, specify a ZIP file in **Image file** that contains only an installation method but no software.
2. For the relevant devices, perform an **Update software** command.

Device configuration

Any modifications to the device configuration are carried out in UniconOS Management Board. They can relate to a particular device, to a particular OU or to all devices.

Modifying device configuration

1. Open the device configuration for the relevant devices. For further information, see Device configuration.
2. Click **Edit**, select the relevant section in the list on the left, and edit the relevant options.
3. Click **Apply** (base configuration or an OU) or **Apply settings** (a device).

On the restart of the relevant Windows devices, the Scout Agent for Windows identifies modifications of the configuration. Normally, the modifications become active after another restart.

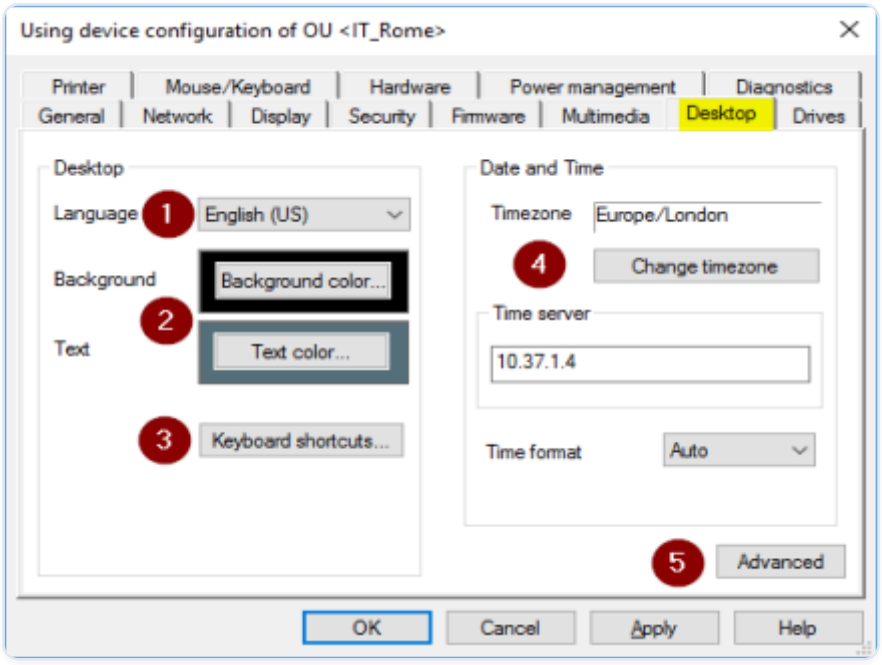
Initial state

Any modifications to the configuration can be undone by resetting the Windows client to initial state by means of the relevant remote command. For Windows Embedded devices, the initial state corresponds to the status right after installing the Windows Embedded image.

Desktop

By using the device configuration in the Scout Console, you can modify the following desktop settings:

- Language
- Time zone



> **Note:**
>
> Language is related to the display of desktop elements but not to text services and input. The Scout Agent for Windows is shown in German if you have selected the German display language. For all other selected languages, it is shown in English.

Screen

By using the device configuration in the Scout Console, you can modify the following screen settings:

- Number of monitors
- Resolution
- Rotation
- Power saving

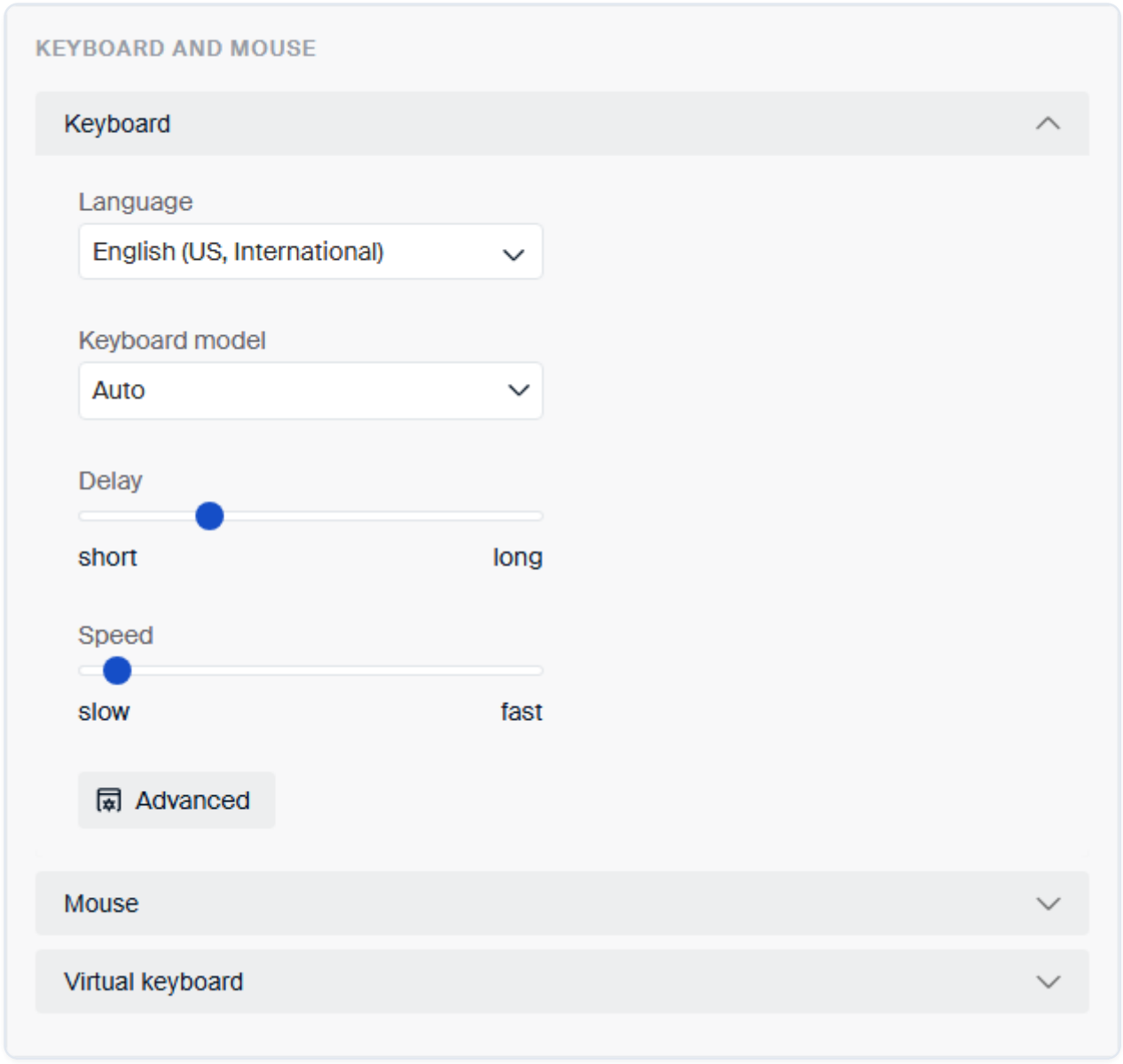
> **Note:**
>
> If the write filter is enabled, modified resolution values are assigned to the Windows client only after the user has logged on.

> For further information, also on multiple monitors, see Screen settings and multiple monitors in the **Scout** guide.

Keyboard

In the **Keyboard and Mouse** section of the device configuration in UniconOS Management Board, you can modify the following keyboard settings:

- Language
- Delay
- Speed



For further information, see Device configuration / Keyboard and Mouse.

Diagnostics

In the Scout Console, in **Device configuration > Diagnostics**, you can enable and disable enhanced debugging on the client.

If the **Debug level** is active, the feature **Device diagnostics** helps you run predefined commands on the client and retrieve a set of configuration and log files to a greater extent than without enhanced debugging.

If you require technical support from Unicon™, enable enhanced debugging before you perform **Device diagnostics.**

Device diagnostics is performed by using an online command, for further information, see Device diagnosis in the **Scout** guide.

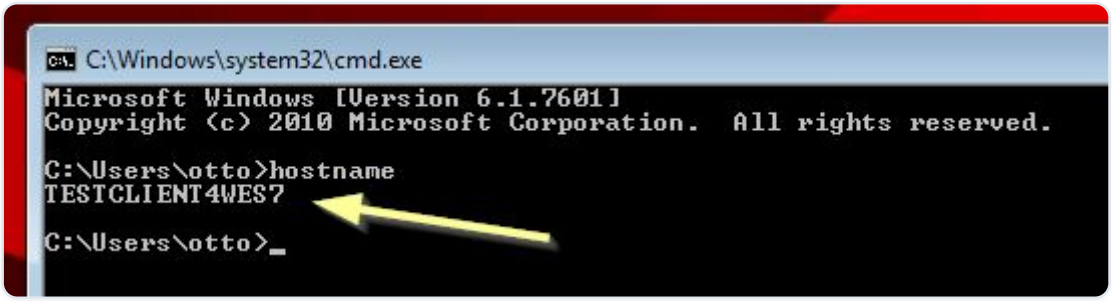
> **Note:**

> Make sure to switch off the **Debug level** after having performed device diagnostics. Otherwise you risk to exceed the memory capacity of the device.

Changing the device name

1. In a device list in UniconOS Management Board, click ⋮ to the right of the device.
2. Choose **Edit** > **Rename**, and enter a new name for the device.
3. Press **Enter**.

The host name on the Windows client is changed accordingly.



Applications

-- for Scout Agent for Windows 1.3 and later versions --

The Scout Agent for Windows can provide the following applications defined in the Scout Console for Windows devices:

- Citrix-Receiver
- Microsoft Remote Desktop Connection
- VMware Horizon Client
- Storefront™
- Browser
- File manager
- Text editor
- Movie player
- On-screen keyboard (English)
- Magnifier

- Custom applications

Installing applications

1. In the Scout Console, for the relevant OU, define the required applications. For further information, see Adding applications in the **Scout** guide.
2. In the Scout Console, configure the application properties. Along with application-specific options the following general options are provided:

OPTION	DESCRIPTION
Start automatically 1 (Selecting or clearing the Start automatically option requires a second restart of the client to become active.)	The application is added to the Windows autostart folder and is automatically started after each system start.
Desktop icon	The application's icon is added to the desktop.

   The Windows system does not provide all of the application options. For further information, see Specifics for applications on Windows clients.
3. In the Scout Console, for the relevant OU, from the context menu, choose **Notifications > Initiate refresh**...

On the next restart of the devices, the application is available on the desktop under **Scout Applications**.

> **Note:**
>
> The applications are provided for all users of the Windows system.

Specifics for applications on Windows devices

Applications on the Windows client that have been defined in the Scout Console can be subject to restrictions or specifics.

GENERAL OPTIONS THAT ARE NOT AVAILABLE FOR WINDOWS DEVICES

- **Application restart**
- Delay for **Start automatically**

- Free parameters
- User-defined desktop icons

MICROSOFT REMOTE DESKTOP CONNECTION

- In the Scout Console, in the application properties **Advanced RPD settings > Advanced**, the **Enable compression** option must be selected.
- pass-through logon is not available
- Passwords sent from Scout are ignored.
- Restricted multi-monitor support

VMWARE HORIZON CLIENT

Scout supports the following options:

- Server and protocol
- Logon with username, password and domain, pass-through logon
- Window size

Further configuration can be done on the related View Connection server.

STOREFRONT™

- Each specified URL receives its own start menu entry (and if defined, its own desktop icon)
- The first start of a StoreFront application might take some time
- The **Advanced options** are not available.

BROWSER

Internet Explorer is used as browser.

The following options are supported:

- Start page
- Enabling kiosk mode

ON-SCREEN KEYBOARD

Requires defining a local Custom application with **Parameter**OSK (no case-sensitivity).

CUSTOM APPLICATION

In the **Parameter** field, make sure to enter only values that exist on the Windows system.

Remote control

Controlling Windows devices remotely means that, in UniconOS Management Board, you schedule and perform commands either on a particular device or on an OU or Dynamic Device Group, see Commands and schedules.

- Remote commands

Remote commands

In UniconOS Management Board, by using the **Actions** menu of a device or OU, you can perform the following remote commands:

- Switch on
- Switch off
- Restart
- Send message
- Update software
- Reset configuration
- User-defined command

The commands can be performed immediately, or scheduled to be run once or periodically.

In addition, to an individual client device, you can apply the features **Device diagnostics** and **Mirroring**. For further information, see Helpdesk features.

For further information on how to apply and schedule remote commands, see Commands and schedules. To provide user-defined commands as ready-to-use commands, see Predefined commands.

Asset management

- Device data in Scout

Device data in Scout

On each system start, the client devices connect to their Scout Server and send their asset data such as hardware properties. The asset data include information on the hardware such as device type, UEFI/BIOS version and flash size, as well as information on the OS version and locally saved licenses.

In UniconOS Management Board, you find these data in the details panel of a device. To compare devices, show the data as columns of a device list, and sort and filter the list by them, for example to find all devices with a flash size of 4 GB or less. You can also export a device list to a CSV file with **Export as CSV**. For further information, see Display devices and Filtering devices.

To use the same columns and filters again, save them in a view, see Individual device list views.

Helpdesk features

- Mirroring client screen
- Device diagnostics

Mirroring client screen

To launch a mirror session, the administrator performs a **Mirror** command on the client (for Scout Agent for Windows 1.3 and later versions).

When you establish a mirror session, all security options apply that have been set for that OU in the **Mirroring** section of the device configuration:

- The password for the mirror session
- **Read-only access**
- **User must confirm within**
- **Encrypt transmission**
- **Allow from UniconOS Management only**
- **Log off on disconnection**

During the mirror session, a system message is shown on the client device which remains in the foreground but can be moved by the user. The system message provides an **Abort** button that allows the user to stop mirroring at any time and to grant only read access to the administrator.

For further information, see Mirroring a device.

> **Note:**

> If the **User must confirm within** option is active, the first attempt to connect is refused. Reconnecting a second time does work.

Device diagnostics

Device diagnostics is a useful feature of UniconOS Management Board. It helps you run predefined commands on the client and retrieve protocol and configuration files from the client for diagnostic purposes. The requested client files support the administrator in error analysis and are required when opening a support ticket.

You can also use this feature to request any definable files from the client.

For further information, see Retrieving diagnostic data.

Implementation Guides

An Implementation Guide is a product manual that describes a dedicated topic or scenario that can be implemented by one or more of our products, including detailed instructions on how to proceed.

We provide Implementation Guides for our eLux® operating systems, our management software Scout Enterprise Management Suite® and also for some additional features.

The documentation refers to the latest product versions.

Persistent User Configurations

This guide describes the Persistent Home feature in UniconOS. When enabled, user-specific configuration data is retained across logoff/logon cycles and device reboots.

By default, Persistent Home is disabled.

Overview

When persistent home is enabled, your personal settings will be saved and selected user configurations remain available across:

- User sign-out and subsequent sign-in
- Device restarts

Persistence is controlled explicitly via configuration (see Configuration).

Supported Applications and Data

Google Chrome

- Persisted data:
  - User profiles
  - Appearance settings (themes, fonts, color schemes)
  - other browser preferences
- Behavior:

- Settings persist across sessions
- Depending on backend configuration, settings may follow the user across devices

Firefox

- Persisted data:
  - Appearance settings (fonts, color schemes
  - Download location/path
  - other browser preferences
- Important:
- See Limitations regarding full profile file support.

ICA (Citrix Workspace App)

- Persisted data:
  - User-specific ICA settings (for example, session-related preferences)
- Continuity behavior:
  - Enables improved session continuity during temporary disruptions in the Citrix Delivery infrastructure.

GNOME Keyring (credentials/secrets)

This option is typically used in combination with browser or ICA persistence.

- Persisted data:
  - Stored credentials
  - Application secrets used by supported applications

Audio

- Persisted data:
  - Each user's list of USB audio devices remembered with **Remember device**. The list is per user and stays on the device — it does not follow the user to another device.
- Requires UniconOS 2609 or later. Earlier versions ignore this option.

- Requires user authentication. When **User authentication** is set to **None**, remembered devices last only for the current session, even with Audio selected.
- UniconOS offers **Remember device** only while **Inform user on changes** (device configuration, **Hardware** > **USB Settings**) is on. When it's off, no notification appears for USB audio devices, so users can't remember one.

UniconOS Management Board also offers **Microsoft Edge** as a persistable application, in addition to the applications described above.

Configuration

Persistent Home can be configured either locally or centrally.

- Local file: `/setup/terminal.ini`
- Central management: Configurable via device profile in Scout

Configuration Parameter

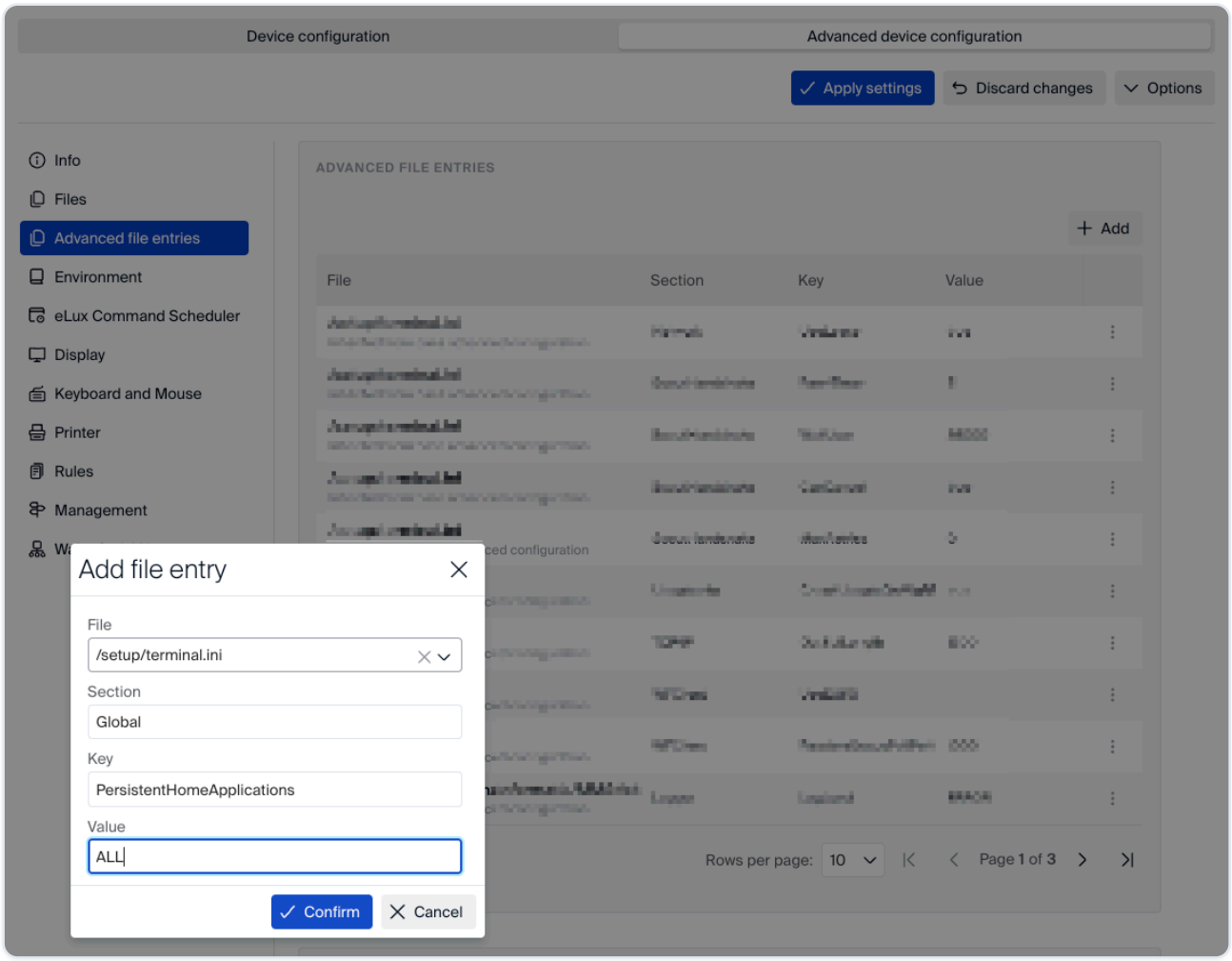
- Use `PersistentHomeApplications` to select which items persist.
- Allowed values:
  - `google-chrome`
  - `firefox`
  - `ica`
  - `gnome-keyring`
  - `audio`
  - `ALL`

Behavior

- Multiple values can be combined as required, separated by commas, for example `google-chrome,audio`.
- ALL enables persistence for all supported applications available in the environment, including Audio.
- gnome-keyring can be combined with:
  - google-chrome
  - ica

- or both

UniconOS Management Board route (from UniconOS Management 2609): go to **Devices** > **Device management** > **Device configuration** (base configuration, an OU or a device) > **Persistent home** > **Enable persistent home configuration** > **Select applications to persist**, then select **Audio** (or any other application). **Persist all applications** includes Audio. The same control is available in **Configuration labels**.



An advanced file entry for `PersistentHomeApplications` overrides the Persistent home section. Add `audio` to that entry too, or remove the entry so the UniconOS Management Board setting takes effect.

Limitations

- Firefox:
  - Customized settings persist
  - Full profile file persistence is not supported in this release

- Persistence behavior may vary depending on:
  - Organizational security policies
  - Infrastructure configuration
  - Central management settings
- A factory reset deletes all Persistent Home data, including every user's remembered audio devices, whether it is started on the device or sent from UniconOS Management Board (**Reset configuration**). **Retain local configuration** does not keep this data. Moving a device to another OU also deletes its Persistent Home data.

Administrators should validate persistence behavior in their specific deployment scenario.

Persistent User Configurations

This guide describes the Persistent Home feature in UniconOS. When enabled, user-specific configuration data is retained across logoff/logon cycles and device reboots.

By default, Persistent Home is disabled.

Overview

When persistent home is enabled, your personal settings will be saved and selected user configurations remain available across:

- User sign-out and subsequent sign-in
- Device restarts

Persistence is controlled explicitly via configuration (see Configuration).

Supported Applications and Data

Google Chrome

- Persisted data:
  - User profiles
  - Appearance settings (themes, fonts, color schemes)
  - other browser preferences
- Behavior:
  - Settings persist across sessions

- Depending on backend configuration, settings may follow the user across devices

Firefox

- Persisted data:
  - Appearance settings (fonts, color schemes
  - Download location/path
  - other browser preferences
- Important:
- See Limitations regarding full profile file support.

ICA (Citrix Workspace App)

- Persisted data:
  - User-specific ICA settings (for example, session-related preferences)
- Continuity behavior:
  - Enables improved session continuity during temporary disruptions in the Citrix Delivery infrastructure.

GNOME Keyring (credentials/secrets)

This option is typically used in combination with browser or ICA persistence.

- Persisted data:
  - Stored credentials
  - Application secrets used by supported applications

Audio

- Persisted data:
  - Each user's list of USB audio devices remembered with **Remember device**. The list is per user and stays on the device — it does not follow the user to another device.
- Requires UniconOS 2609 or later. Earlier versions ignore this option.
- Requires user authentication. When **User authentication** is set to **None**, remembered devices last only for the current session, even with Audio selected.

- UniconOS offers **Remember device** only while **Inform user on changes** (device configuration, **Hardware** > **USB Settings**) is on. When it's off, no notification appears for USB audio devices, so users can't remember one.

UniconOS Management Board also offers **Microsoft Edge** as a persistable application, in addition to the applications described above.

Configuration

Persistent Home can be configured either locally or centrally.

- Local file: `/setup/terminal.ini`
- Central management: Configurable via device profile in Scout

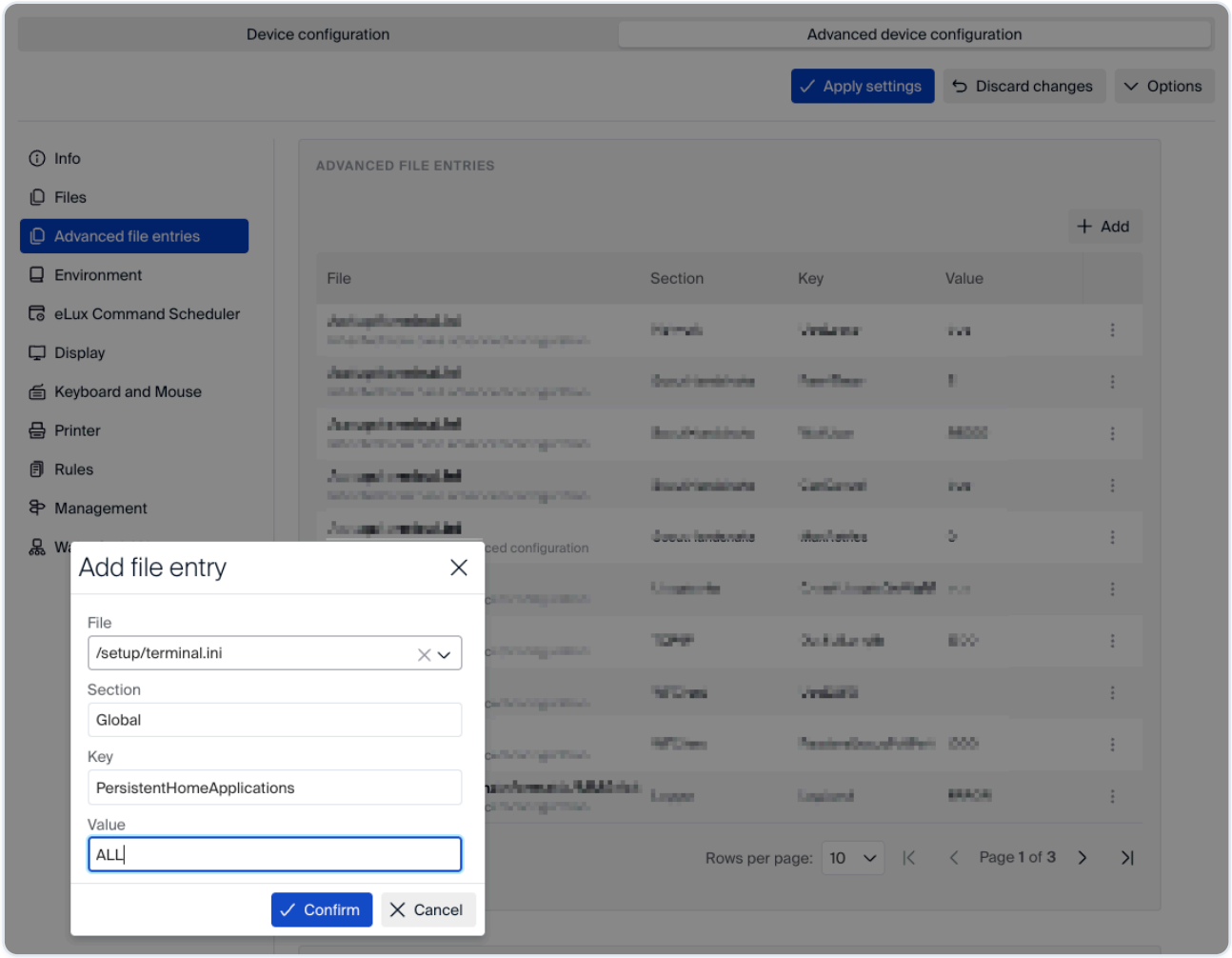
Configuration Parameter

- Use `PersistentHomeApplications` to select which items persist.
- Allowed values:
  - `google-chrome`
  - `firefox`
  - `ica`
  - `gnome-keyring`
  - `audio`
  - `ALL`

Behavior

- Multiple values can be combined as required, separated by commas, for example `google-chrome,audio`.
- ALL enables persistence for all supported applications available in the environment, including Audio.
- gnome-keyring can be combined with:
  - google-chrome
  - ica
  - or both

UniconOS Management Board route (from UniconOS Management 2609): go to **Devices** > **Device management** > **Device configuration** (base configuration, an OU or a device) > **Persistent home** > **Enable persistent home configuration** > **Select applications to persist**, then select **Audio** (or any other application). **Persist all applications** includes Audio. The same control is available in **Configuration labels**.



An advanced file entry for `PersistentHomeApplications` overrides the Persistent home section. Add `audio` to that entry too, or remove the entry so the UniconOS Management Board setting takes effect.

Limitations

- Firefox:
  - Customized settings persist
  - Full profile file persistence is not supported in this release
- Persistence behavior may vary depending on:

    - Organizational security policies
    - Infrastructure configuration
    - Central management settings

- A factory reset deletes all Persistent Home data, including every user's remembered audio devices, whether it is started on the device or sent from UniconOS Management Board (**Reset configuration**). **Retain local configuration** does not keep this data. Moving a device to another OU also deletes its Persistent Home data.

Administrators should validate persistence behavior in their specific deployment scenario.

AD Credentials Pass-Through for Google Chrome

Overview

eLux supports **Active Directory (AD) credential pass-through for Google Chrome**, enabling seamless access to internal web resources without repeated authentication prompts.

> **Important:**

> This SSO Support option cannot be enabled together with Chrome Enterprise Premium (CEP). CEP already provides single sign-on for Citrix Secure Access™. If CEP is enabled, disable this SSO Support option in the image definition.

Key Capabilities

- **Single sign-on (SSO)** After logging in to eLux with AD credentials, Google Chrome automatically reuses the same Kerberos-based authentication context (Kerberos TGT).
- **Policy-Controlled Experience** Chrome enforces centrally defined policies, ensuring transparent and compliant access to corporate web services.

Prerequisites

Ensure the following conditions are met before configuration:

- AD authentication is enabled and functional in Scout.
- The user successfully logs in to eLux with AD credentials.
- Google Chrome is installed on the eLux endpoint.
- You have sufficient permissions to modify the device profile in Scout.

Configuration (Scout)

1. Open the relevant device profile in **Scout**.
2. Navigate to **Applications → Google Chrome → Free parameters**.
3. Add the following parameter and save the configuration: `AllowSSOLists=example.com`
   Notes:
   - "example.com" is an example and may be exchanged with your domains.
   - The parameter accepts a comma separated list of domains.
   - The parameter accepts Google Chrome policy wildcards (e.g. *.example.com)
4. Save and deploy the profile to the target devices.

End-User Experience

1. The user logs in to eLux using their AD credentials.
2. Google Chrome is launched (manually or via autostart).
3. When accessing any domain listed in **AllowSSOLists**, Chrome performs automatic authentication without prompting for credentials.

Troubleshooting

Google Chrome still prompts for a password

- Verify the user logged in with an AD account.
- Confirm the destination site is included in **AllowSSOLists**.
- Redeploy the Scout profile if changes were made recently.

POLICY NOT APPLIED

- In Chrome, enter `chrome://policy` and check that **AuthNegotiateDelegateAllowlist** shows the expected entries.

Need Help?

Contact your system administrator or eLux support for further assistance.

Entra ID Authentication at eLux Login

This guide describes how to configure Microsoft Entra ID authentication at the eLux® login screen using FIDO2 (a passwordless sign-in standard that uses a security key), with single sign-on (SSO) into Citrix Workspace™ app (CWA). It applies to eLux 7.2609 with Citrix Workspace app 26.08.

At login, eLux opens Chrome to handle Entra ID sign-in instead of showing the built-in login screen. Chrome persists the Entra ID sign-in session, so CWA reuses it afterward for both the store and the VDA (Virtual Delivery Agent, the Citrix® component that hosts the session) — the user authenticates only once.

Prerequisites

The eLux image must include the following before you enable this feature:

REQUIREMENT	WHY
`google_chrome` EPM (eLux software package), including the `google_chrome_sso_browser` FPM (a feature within it)	Provides `/usr/bin/unicon-sso-browser`, the wrapper that owns the shared browser profile. Verified with Chrome 149 and 153.
`ica` EPM — CWA 2608 (tested with build 26.08.0.191)	Supplies the store and VDA legs. The Early Access Release (EAR) build 26.08.0.153 is not usable: its shared-browser login window renders nothing and closes.
`systemlibs_libxml2_utils` (provides `xmllint`)	`/etc/elux/eluxd.d/ica` validates `AuthManConfig.xml` with `xmllint`. If the binary is missing, the file is treated as invalid and all of `sessions.ini [ICADefaults]` is silently skipped, so `FIDO2Enabled` and `AADSSOEnabled` never reach CWA. The 2609 recovery base doesn't include it. With CWA 2608 GA, **Software** in UniconOS Management Board adds it as a dependency of the `ica` package when it solves the dependencies of the image. If it's missing from the image definition file (the `.idf` file that lists which packages the image installs), add it.
Gateway or StoreFront CA certificate in `/setup/cacerts`	The authentication step now runs in Chrome's own trust store instead of CWA's. An internal CA that CWA tolerated will fail with `ERR_CERT_AUTHORITY_INVALID` once this feature is enabled.

Configuration

Configure this feature as Scout advanced INI entries on the OU (organizational unit). All nine entries are required, and the device must restart afterward — the logon and lock screens decide whether to use Chrome for sign-in at startup.

FILE	SECTION	KEY	VALUE	WHAT IT CONTROLS
`terminal.ini`	`[Security]`	`SsoBrowserEnabled`	`true`	The eLux side: logon screen, lock screen, and logout hook. Defaults to `false`. Does not affect CWA by itself.
`terminal.ini`	`[Security]`	`BrowserTrustedUris`	`https://accounts.<store-domain>;https://accounts.cloud.com;https://login.microsoftonline.com`	Origins Chrome may hand off to without prompting. Must include the identity provider's own host — the device cannot derive this from the store URL.
`terminal.ini`	`[Browser]`	`SuppressPrompts`	`true`	Turns on prompt suppression in Chrome, Edge and Firefox, see Browser Prompts.
`sessions.ini`	`[ICADefaults]`	`FIDO2Enabled`	`true`	Routes the store login through the shared browser.

FILE	SECTION	KEY	VALUE	WHAT IT CONTROLS
`sessions.ini`	`[ICADefaults]`	`FIDO2AuthBrowser`	`unicon-sso-browser`	Specifies which browser to use.
`sessions.ini`	`[ICADefaults]`	`FIDO2AuthTimeout`	`180`	The default 60 seconds is not enough for an interactive sign-in, especially with a security key.
`sessions.ini`	`[ICADefaults]`	`AADSSOEnabled`	`true`	Routes the VDA login through the shared browser. Verified on-device: this key already defaults to `true`; only set it if you need to turn the behavior off (`false`).
`sessions.ini`	`[ICADefaults]`	`AADSSOWithFido2AuthenticationEnabled`	`true`	Pairs the VDA login with the FIDO2 browser.
`sessions.ini`	`[Chrome]`	`ExtraFlags`	`--hide-crash-restore-bubble`	Suppresses Chrome's "Restore pages?" prompt after a session ends.

With UniconOS Management 2611 and later, you can set `SuppressPrompts` and `BrowserTrustedUris` in UniconOS Management Board instead, with **Suppress browser prompts** and **Trusted browser origins** in the **User authentication** section of the device

configuration, see Browser prompts. If you do, delete the advanced file entries for these two keys, because they override the settings.

Turning off the feature

Set these three keys to `false` to disable it:

SETTING	VALUE
`terminal.ini` `[Security]` `SsoBrowserEnabled`	`false`
`sessions.ini` `[ICADefaults]` `FIDO2Enabled`	`false`
`sessions.ini` `[ICADefaults]` `AADSSOEnabled`	`false`

Setting only `SsoBrowserEnabled` to `false` turns off the eLux login and lock screen behavior, but CWA keeps using the shared browser if `FIDO2Enabled` and `AADSSOEnabled` are still `true`.

Boot-to-VDI with eLux Loginmanager

Starting in eLux® 7.2609.0, a boot-to-VDI kiosk uses the eLux Loginmanager (the component that handles sign-in at device startup) as its single authentication entry point, instead of handing the user straight to Citrix Workspace™ app (CWA) at launch. The device boots to one eLux sign-in screen. After the user authenticates once, eLux launches the assigned Citrix desktop or app full-screen and locked — the local eLux desktop is never exposed before, during, or after the session. Boot-to-VDI is one of several scenarios delivered under Boot-to-X.

What the user sees

ACTION	RESULT
Power on	The eLux sign-in screen appears; no local desktop is shown.
Authenticate once (badge, smartcard, AD, or identity provider)	Loading holds until the Citrix® desktop is visually ready, then it fills the screen and locks.

ACTION	RESULT
End the session (sign out, disconnect, or tap out)	The device returns to the eLux sign-in screen; no state carries over to the next user.
Tap the same Imprivata badge again	The session locks instead of closing — tapping the same badge again resumes it without re-authenticating. If the Imprivata policy logs users off when they leave the device, the tap logs the user off instead, see Lock or Log Off When Users Leave the Device.
Tap a different Imprivata badge	The user switches: the previous session is released and the new user's desktop launches.
The launch fails or the session is lost	A recoverable error panel appears with **Retry** and **Back** — never a black or frozen screen.

With Imprivata XenDesktop® automation, eLux reads the user's desktops from the StoreFront™ stores in the Imprivata computer policy. A user with several desktops picks one in the Desktop Chooser before the desktop launches.

Requirements

- eLux 7.2609.0 or later on the endpoint.
- Citrix Workspace app for Linux 2608 or later.
- A published desktop or app assigned to the user, with StoreFront or Citrix DaaS reachable from the endpoint.
- An eLux authentication method configured: Imprivata, Active Directory (username and password), smartcard, or a federated or cloud identity provider (Entra ID, OpenID, SAML, Okta, or PureAuth).

Configuration

From UniconOS Management 2611, you set up Boot-to-VDI in UniconOS Management Board, in the **User authentication** section of the device configuration:

1. Select an authentication type that supports Boot-to-VDI.
2. Set **Start scenario** to **Boot to VDI**.
3. Enter the **Store** and the **Resource**.

4. Restart the devices after they get the new configuration.

For details, see Start scenario in the **UniconOS Management Board** guide.

Scout sends these entries to the device:

#	SETTING	WHERE	VALUE
1	Boot-to-VDI scenario (`ScenarioType`)	`/setup/terminal.ini`, `[Security]`	`1` (`0` = eLux desktop)
2	Store to enumerate (`ScenarioStore`)	`/setup/terminal.ini`, `[Security]`	The StoreFront store that hosts the resource
3	Resource to launch (`ScenarioResource`)	`/setup/terminal.ini`, `[Security]`	The enumerated published desktop or app name
4	Boot-to-VDI reporting (`BootToVDIEnabled`)	`/setup/ica/wfclient.ini`, `[WFClient]`	`True`, enables the reveal-on-ready hand-off

With Imprivata XenDesktop automation, leave **Resource** (`ScenarioResource`) empty, see VDI Access Policy.

With Scout 2609 or earlier, add the same entries manually as advanced file entries. eLux reads the `Scenario*` keys only in `[Security]`. In `[Global]`, they have no effect, and the device starts the eLux desktop.

Smart Card Actions Configuration

Goal

Configure Smart Card Actions on eLux devices to control system behavior when a smart card is inserted or removed. This includes session locking, logoff, user switching, and execution of custom scripts.

Prerequisites

Before you begin, ensure the following:

- You have access to **Scout** with permission to edit device profiles.
- The target device is assigned to the correct profile.
- Smart card hardware is installed and functioning.
- Any required Bash scripts are prepared and available for upload.

Configuration Overview

The Smart Card Actions feature provides seven configuration parameters. Configuration must be done via Advanced File Entries:

- **Scout → Device Profile → Advanced device configuration → Advanced file entries**

The relevant sections are:

```
[Hotkeys]

PauseAction



[SmartCard]

OnScInsert

OnScRemoval

ScInsertScript

ScRemoveScript

PauseActionOnDifferentCardInserted

PauseActionTimeoutSeconds
```

Configuration Details

The following parameters control system behavior when a smart card is inserted, paused or removed.

PauseAction

Defines the modifier key used to trigger the pause action.

- **Default value:** `<shift>`
- **Valid values:** `<shift>`, `<ctrl>`, `<alt>`

> This setting does not distinguish between left and right modifier keys.

OnScInsert

Defines the action performed when a smart card is inserted.

- **Default value:** `none`
- **Valid values:** `none`, `runscript`
- **Requirement:** Value must be lowercase.

VALUE	BEHAVIOR
`none`	No action is performed.
`runscript`	Executes the script defined in `ScInsertScript`.

OnScRemoval

Defines the action performed when a smart card is removed.

- **Default value:** `none`
- **Valid values:** `none`, `lock`, `logoff`, `runscript`
- **Requirement:** Value must be lowercase.

VALUE	BEHAVIOR
`none`	No action is performed.
`lock`	Locks the current eLux session.
`logoff`	Logs off the current eLux session.

VALUE	BEHAVIOR
`runscript`	Executes the script defined in `ScRemoveScript`.

ScInsertScript

Specifies the script executed when a smart card is inserted.

- **Default value:** `empty`
- **Valid values:** Full path to a Bash script on the device.

> The file must exist on the device and match the configured path exactly.

ScRemoveScript

Specifies the script executed when a smart card is removed.

- **Default value:** `empty`
- **Valid values:** Full path to a Bash script on the device.

> The file must exist on the device and match the configured path exactly.

PauseActionOnDifferentCardInserted

Defines system behavior when a different smart card is inserted during an active session.

- **Default value:** `switchuser`
- **Valid values:** `switchuser`, `ignore`
- **Requirement:** Value must be lowercase.

VALUE	BEHAVIOR
`switchuser`	Logs off the current session and displays the Smart Card login screen.
`ignore`	Continues the current session without interruption.

PauseActionTimeoutSeconds

Defines the timeout (in seconds) for the pause action.

- **Default value:** `30`
- **Valid values:** Numeric (5–300 seconds)

Behavior outside valid range:

- Values below `5` are automatically adjusted to `5`.
- Values above `300` are automatically adjusted to `300`.

Script Files

Smart Card Actions support custom **Bash scripts** that can be executed on card insertion or removal.

Supported Script Type

- Only **Bash (`.sh`) scripts** are supported.
- Scripts must be uploaded in **Scout** under:

Advanced device configuration → Files

File Location Requirements

The **destination path** configured during file upload must exactly match the value specified in:

- `ScInsertScript`
- `ScRemoveScript`

If the configured path and the uploaded file location do not match, the script will not execute.

> Ensure the script has the correct permissions and is accessible on the device.

Imprivata Guide

This documentation provides guidance on the integration of **Imprivata authentication** with **eLux**, enabling secure, streamlined, and user-friendly access to applications and desktops in data-sensitive environments, with a particular focus on healthcare.

Modern hospitals and care facilities rely on fast, reliable, and secure authentication workflows to reduce login delays and minimize administrative burden for clinicians. With this integration, eLux

acts as the endpoint operating system for thin clients and workstations, while Imprivata serves as the **central authentication backend**. This combination delivers a strong, enterprise-grade solution for **Single Sign-On (SSO)** and **proximity card (Tap&Go)** access, tailored to the unique needs of healthcare IT.

Purpose of the Integration

The integration allows clinicians to:

- **Log in quickly** using Imprivata-supported methods such as username/password or proximity cards.
- **Automatically connect** to Citrix StoreFront sessions using Single Sign-On, reducing the need for repeated credentials input.
- **Maintain security and compliance** while ensuring resilience against login errors (e.g., incorrect PIN/password entries).

By linking Imprivata's authentication backend with eLux, users benefit from:

- **Improved clinical workflow efficiency:** Caregivers spend less time on login screens and more time with patients.
- **Reduced support overhead:** Centralized authentication policies simplify IT management.
- **Compliance with healthcare security standards:** Multi-factor authentication and controlled session access safeguard sensitive patient data.
- **Resilience and continuity:** Robust error handling and session management designed for 24/7 healthcare operations.

Target Audience

This documentation is intended for:

- **IT administrators** managing hospital endpoints and Citrix environments.
- **System integrators** deploying Imprivata in healthcare institutions alongside eLux OS.

It assumes familiarity with:

- Citrix StoreFront / Virtual Apps and Desktops
- Imprivata ProveID appliance configuration

Workflows

eLux supports multiple authentication workflows, which can be applied individually or in combination depending on organizational requirements.

Authentication Only Provides secure authentication through the Imprivata appliance at device login.

After successful authentication, the user receives access to the standard eLux desktop with all configured applications.

Fast User Switching (FUS) Enables immediate access to a predefined desktop or application after login, reducing delays and supporting quick role changes in clinical environments.

EPCS Reauthentication Supports clinical workflows that require reauthentication within target applications running on VDI, leveraging virtual channel integration.

XenDesktop® Automation with Desktop Chooser Each user who taps in gets their own Citrix® desktop, so that users with different desktops can share one device. At every login, eLux reads the user's desktops from the StoreFront™ stores in the Imprivata computer policy. A user with several desktops picks one in the desktop chooser. This workflow needs boot-to-VDI and can't be combined with Fast User Switching. See XenDesktop Automation and the Desktop Chooser.

Log Off When Users Leave the Device On a shared device, a tap-out with the card, the hot key or inactivity can log the user off instead of locking the device. The Citrix session is then disconnected: the next person at the device can't reach it, but the user can reconnect to it later. The Imprivata policy decides between locking and logging off. See Lock or Log Off When Users Leave the Device.

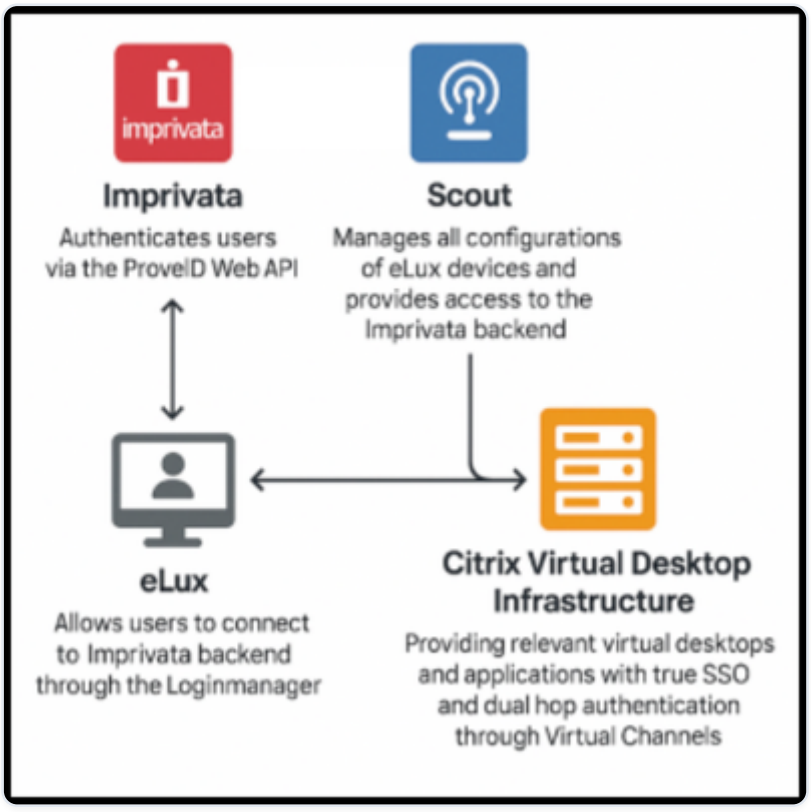
Web Applications at Sign-in Right after an Imprivata sign-in, Chrome opens the web applications of the user, and the Imprivata browser extension signs the user in to them. Each user gets the web applications from their own Imprivata user policy, also on a shared device. This workflow needs Imprivata EAM 26.3 or later. See Opening web applications automatically.

Combined Workflows Workflows can also be combined. For example, **Kiosk Mode** can be paired with **Fast User Switching** to enable rapid handovers while maintaining direct access to a core application.

Architecture

Secure authentication with Imprivata in eLux is achieved through the tight integration of four system components:

- **Imprivata** – Authenticates users via the ProveID Web API.
- **Scout** – Manages eLux device configurations and provides connectivity to the Imprivata backend.
- **eLux** – Connects users to Imprivata through the Login Manager and serves as the fundamental operating system.
- **Citrix Virtual Desktop Infrastructure (VDI)** – Delivers desktops and applications with full SSO and dual-hop authentication via Virtual Channels.



eLux Usage

Login Manager

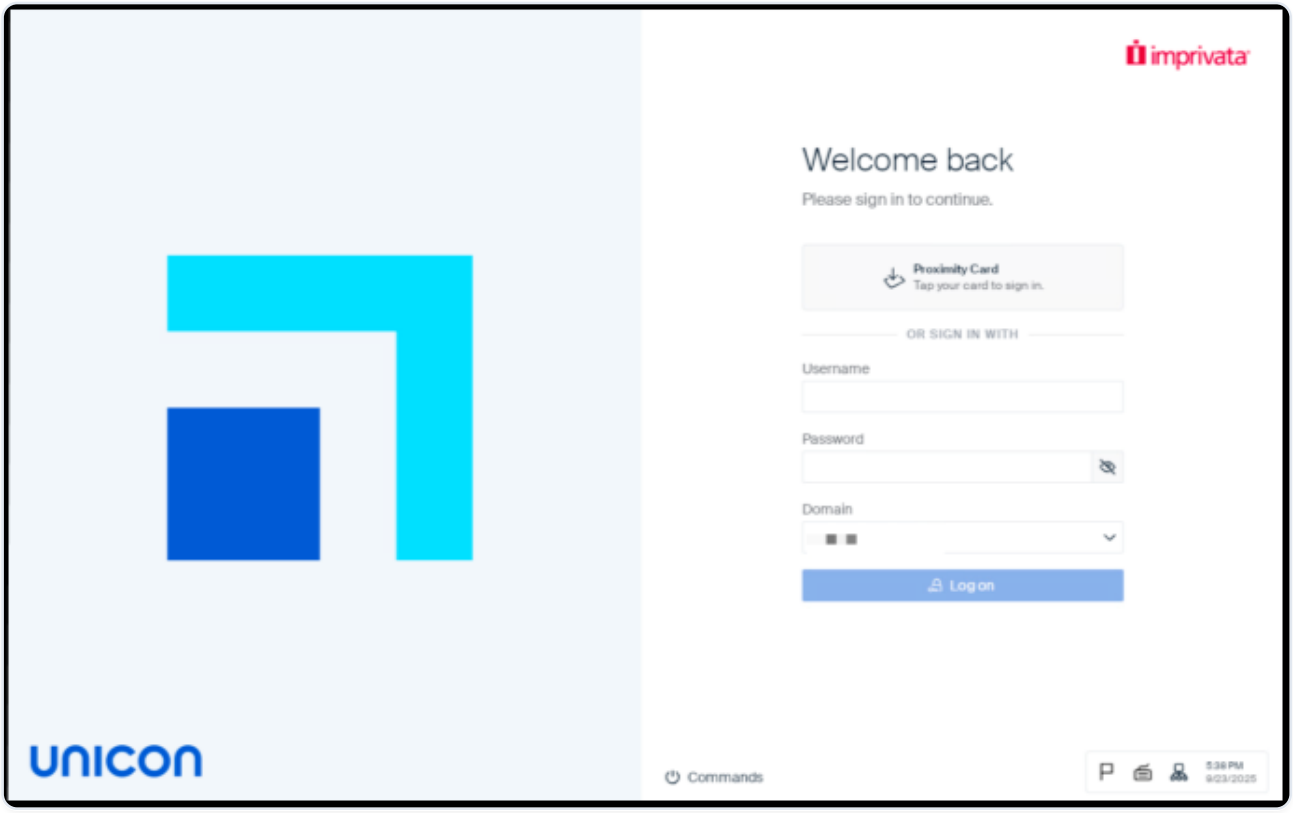
When an eLux device is configured to use **Imprivata** as the authentication backend, the Login Manager displays the **Imprivata logo** after startup.

If a supported card reader is connected, the interface will present a **Proximity Card field**, indicating that **Tap & Go** authentication is available.

The specific login methods offered to the user depend on the configuration defined in the Imprivata backend and may include:

- Username and Password

- Proximity Card with Password
- Proximity Card with PIN
- Proximity Card only



Lock Screen

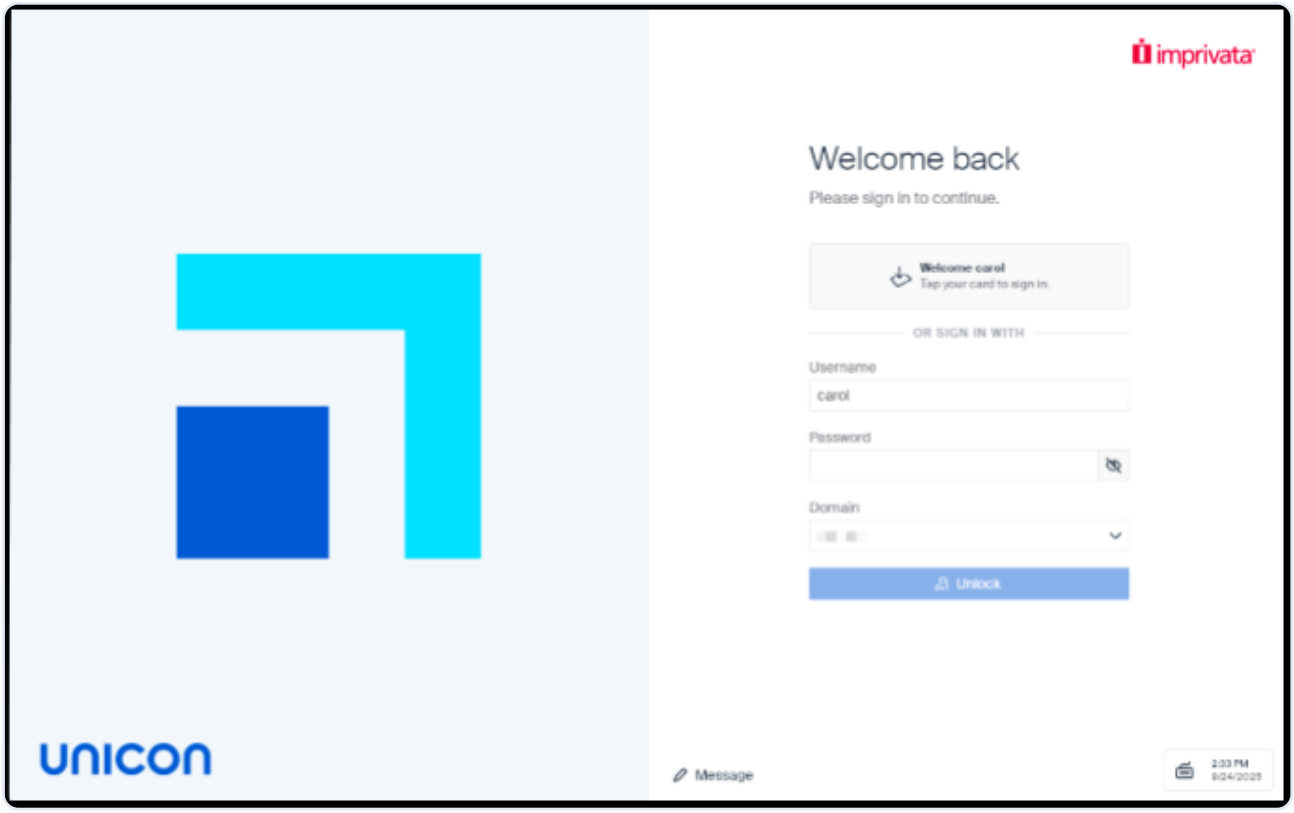
An eLux device can be locked securely using different methods, depending on the system configuration:

- **Card tap** – Quickly lock the system by tapping the proximity card.
- **Start Menu** – Select *Lock* from the eLux Start Menu.
- **Keyboard shortcut** – Use the configured key combination.
- **Inactivity timeout** – The device locks automatically after a defined period of no user interaction.

Once the lock screen is active:

- The currently signed-in user can unlock the session using one of the configured authentication methods.
- Alternatively, another user may log in, which will end the active session of the first user.

From eLux 7.2611.0, the Imprivata policy can log the user off instead of locking the device. A card tap, the keyboard shortcut and the inactivity timeout then log the user off, and only **Lock** in the Start menu locks the device. The Imprivata policy also sets the hot key and the inactivity time. See Lock or Log Off When Users Leave the Device.



Imprivata Card and PIN Enrollment

Goal

Enable users to enroll additional authentication modalities (smart card and PIN) directly during login on eLux devices integrated with Imprivata. This ensures that unknown cards can be registered and required PINs can be set without separate administrative steps.

Feature Overview

eLux supports dynamic enrollment of the following Imprivata authentication modalities during login:

- **Card Enrollment**
  - When a user taps a smart card that is not yet registered in the Imprivata backend, the card can be enrolled during the login process.
- **PIN Enrollment**

- If the PIN modality is configured in the Imprivata backend but no PIN has been enrolled yet, the user is required to enroll a PIN during login.

No additional client-side configuration is required in eLux.

Prerequisites

Before using card or PIN enrollment, ensure the following:

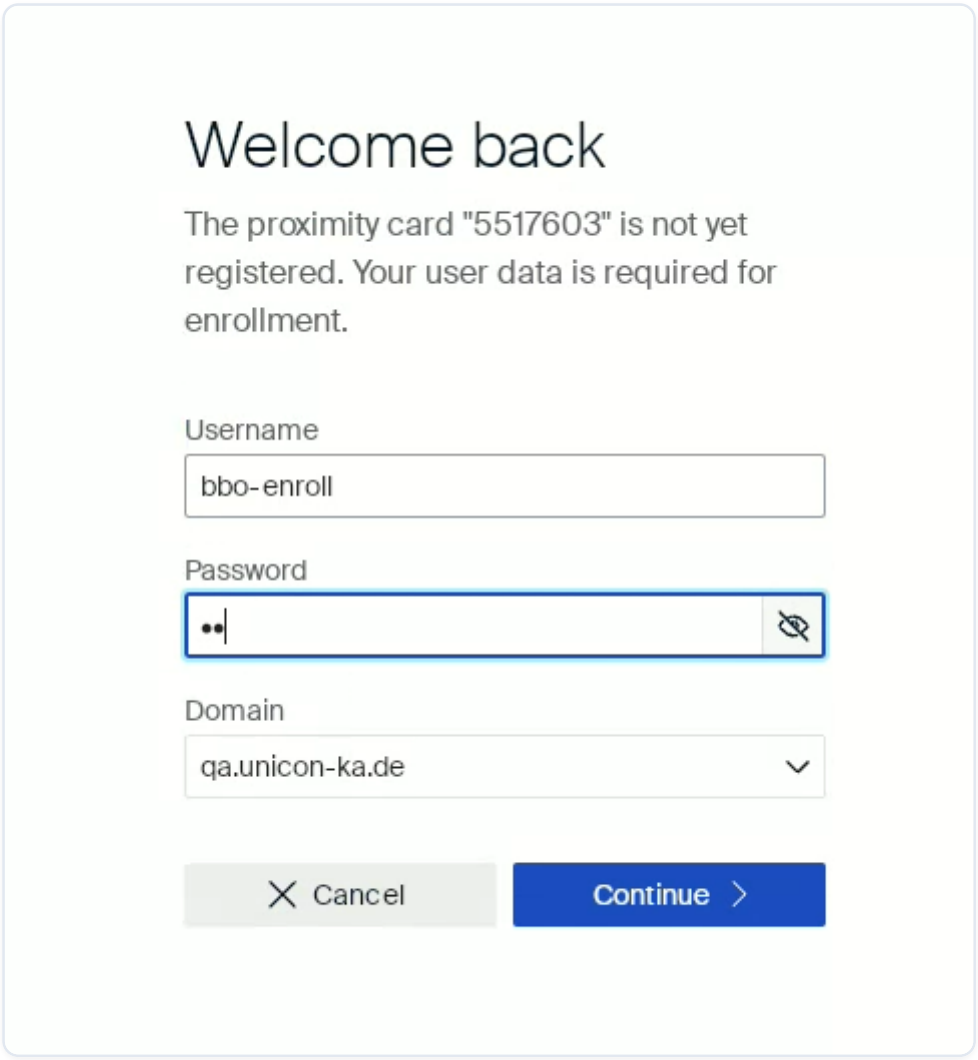
- eLux device is integrated with Imprivata.
- The Imprivata backend is reachable from the device.
- The user account exists in the Imprivata backend.
- For PIN enrollment:
  - The **PIN modality** is enabled for the user in the Imprivata backend.
- For card enrollment:
  - Card enrollment is allowed in the Imprivata backend configuration.

Authentication Flow with Enrollment

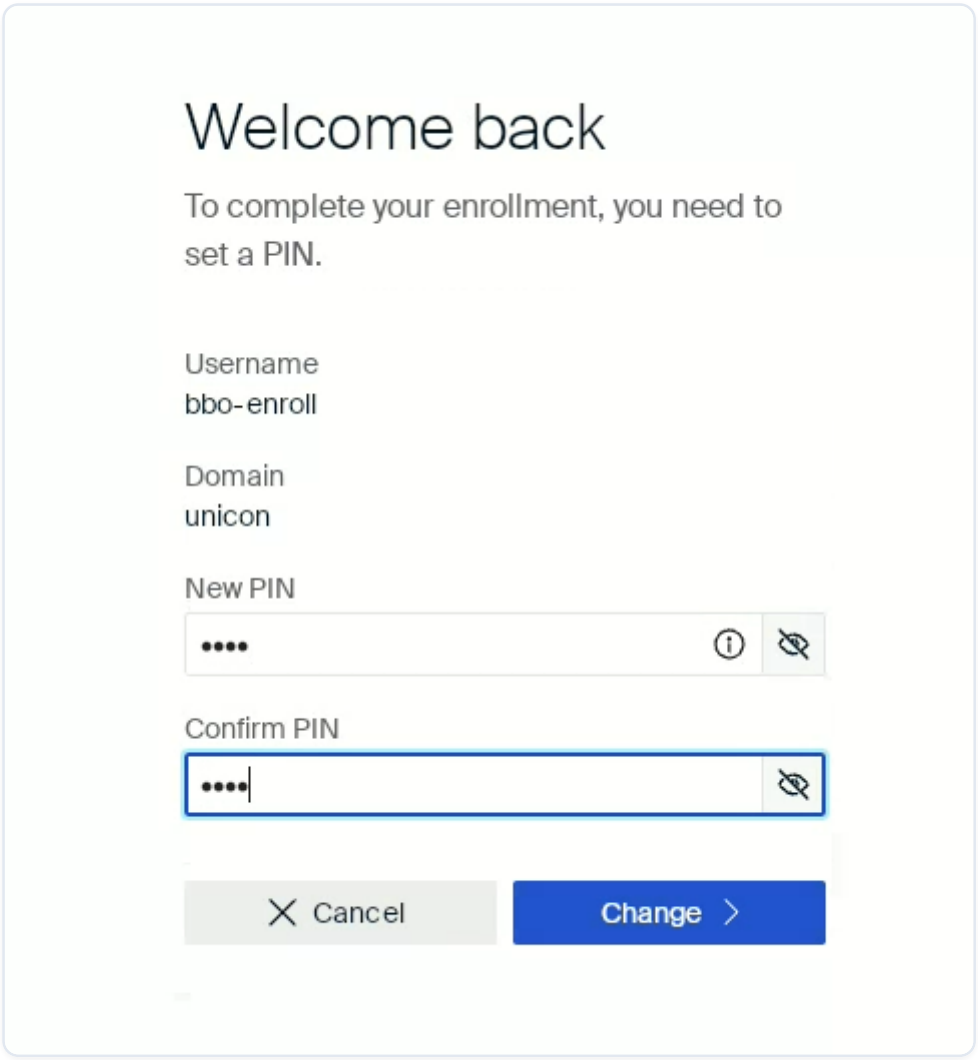
The following describes the authentication flow when both card and PIN enrollment are required:

1. The user taps an unknown smart card.
2. eLux detects that the card is not registered in the Imprivata backend.
3. The user is prompted to enroll the card.
4. The user provides:
   - Username
   - Password
   - Domain
5. If credentials are valid, the card is enrolled and associated with the user account.
6. If the backend requires PIN enrollment, the user is prompted to set a new PIN.
7. After successful PIN enrollment, the user is logged in.

Example card enrollment UI:



Example pin enrollment UI:



Configuration

This section describes relevant configuration aspects on both eLux and the Imprivata backend.

eLux Configuration

No additional configuration parameters are required. The enrollment functionality is enabled by default and cannot be disabled on the client side.

Imprivata Backend Configuration

CARD ENROLLMENT

- Card enrollment must be permitted in the Imprivata backend.
- If card enrollment is disabled:
  - The eLux client displays an error message when an unknown card is tapped.
  - Enrollment cannot proceed.

PIN ENROLLMENT

- The **PIN modality** must be configured for the user.
- If configured and no PIN is enrolled:
  - The user is required to complete PIN enrollment before login.
- If the PIN modality is not configured:
  - The user is not prompted for PIN enrollment.

Known Limitations

- PIN enrollment is triggered immediately if:
  - The PIN modality is configured in the Imprivata backend, and
  - No PIN has been enrolled yet.
- This applies even if the user logs in using username and password instead of a smart card.

> **Example:**

> A user logs in with username and password. If the PIN modality is configured but no PIN exists yet, the user must enroll a PIN before access is granted.

Imprivata Card and PIN Enrollment

Goal

Enable users to enroll additional authentication modalities (smart card and PIN) directly during login on eLux devices integrated with Imprivata. This ensures that unknown cards can be registered and required PINs can be set without separate administrative steps.

Feature Overview

eLux supports dynamic enrollment of the following Imprivata authentication modalities during login:

- **Card Enrollment**
  - When a user taps a smart card that is not yet registered in the Imprivata backend, the card can be enrolled during the login process.
- **PIN Enrollment**

- If the PIN modality is configured in the Imprivata backend but no PIN has been enrolled yet, the user is required to enroll a PIN during login.

No additional client-side configuration is required in eLux.

Prerequisites

Before using card or PIN enrollment, ensure the following:

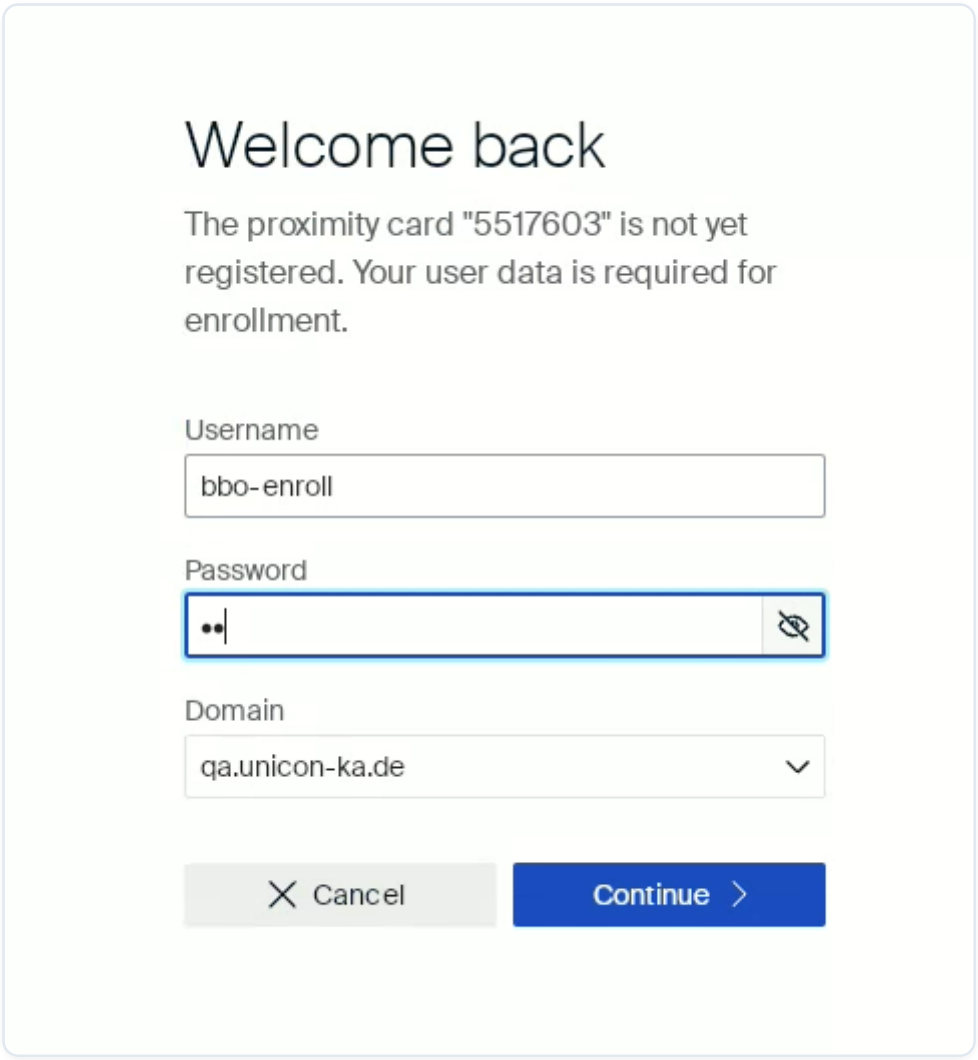
- eLux device is integrated with Imprivata.
- The Imprivata backend is reachable from the device.
- The user account exists in the Imprivata backend.
- For PIN enrollment:
  - The **PIN modality** is enabled for the user in the Imprivata backend.
- For card enrollment:
  - Card enrollment is allowed in the Imprivata backend configuration.

Authentication Flow with Enrollment

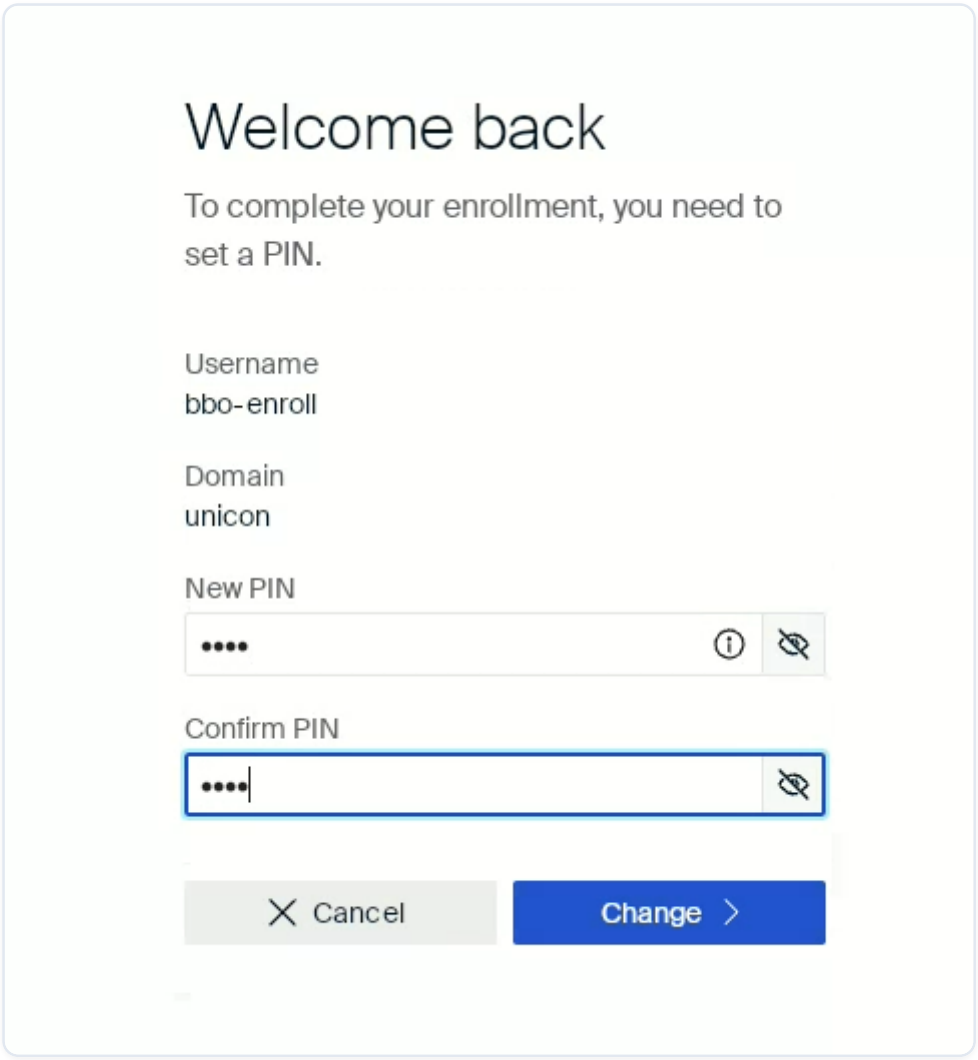
The following describes the authentication flow when both card and PIN enrollment are required:

1. The user taps an unknown smart card.
2. eLux detects that the card is not registered in the Imprivata backend.
3. The user is prompted to enroll the card.
4. The user provides:
   - Username
   - Password
   - Domain
5. If credentials are valid, the card is enrolled and associated with the user account.
6. If the backend requires PIN enrollment, the user is prompted to set a new PIN.
7. After successful PIN enrollment, the user is logged in.

Example card enrollment UI:



Example pin enrollment UI:



Configuration

This section describes relevant configuration aspects on both eLux and the Imprivata backend.

ELUX CONFIGURATION

No additional configuration parameters are required. The enrollment functionality is enabled by default and cannot be disabled on the client side.

IMPRIVATA BACKEND CONFIGURATION

Card Enrollment

- Card enrollment must be permitted in the Imprivata backend.
- If card enrollment is disabled:
  - The eLux client displays an error message when an unknown card is tapped.
  - Enrollment cannot proceed.

PIN Enrollment

- The **PIN modality** must be configured for the user.

- If configured and no PIN is enrolled:
  - The user is required to complete PIN enrollment before login.
- If the PIN modality is not configured:
  - The user is not prompted for PIN enrollment.

Known Limitations

- PIN enrollment is triggered immediately if:
  - The PIN modality is configured in the Imprivata backend, and
  - No PIN has been enrolled yet.
- This applies even if the user logs in using username and password instead of a smart card.

> **Example:**

> A user logs in with username and password. If the PIN modality is configured but no PIN exists yet, the user must enroll a PIN before access is granted.

Default Domain for Login and Lock Screens

By default, the eLux login and lock screens preselect the first domain in the domain picker, even when more domains are available.

Users then have to manually switch domains on every login.

An administrator can configure a **default domain** that is automatically preselected instead, for AD (password), AD with smartcard, and Imprivata login and lock screens:

- **Preselection** — The configured domain is preselected on the login screen, unlock screen, and Imprivata picker, instead of the first domain in the list.
- **User override** — Users can still pick a different domain for their login attempt; the choice isn't overwritten while they're interacting with the screen.
- **Fallback** — If the configured domain is empty, invalid, or no longer available, AD flows fall back to the configured default Kerberos realm, or otherwise a valid domain from the list — login is never blocked.

No additional client-side configuration is required beyond the setting below.

Configuration

Set the default domain in `/setup/terminal.ini`, section `Security`, key `DefaultDomain`, pushed via Scout Advanced File Entries:

File	`/setup/terminal.ini`
Section	`Security`
Key	`DefaultDomain`
Value	The domain to preselect

Enter the domain as it appears in the domain list — either the Kerberos realm (e.g. `DEV.unicon-ka.de`) or the display name (e.g. `DEV`).

Matching is case-insensitive, and the value is canonicalized against the live domain list automatically.

The selected domain is resolved to the correct Kerberos realm before authenticating. For smartcard login, the realm is inferred from the certificate first and only falls back to the selected domain if needed.

The change takes effect the next time the login or lock screen appears.

Good to Know

- On the main unlock screen, AD users re-authenticate as the locked-in user — there's no domain picker.
- If the configured value doesn't match any domain in the list, the device log records `DefaultDomain '<value>' does not match any available domain`.

Grace Period for Imprivata Authentication

Grace period is an Imprivata setting that lets a user re-authenticate with just a proximity card tap, no PIN or password, for a configurable time after their last multi-factor login. It applies to any Imprivata policy that pairs a proximity card with a PIN or password as the second authentication factor.

- **Initial login** — The user taps their card, enters the PIN or password, and the session opens. The grace-period timer starts at that point.

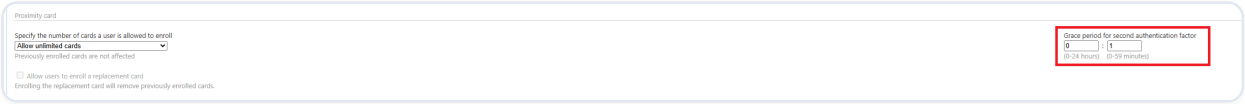
- **Re-authentication within the grace period** — Tapping the card alone unlocks the screen or opens a new session, with no PIN or password prompt.
- **Re-authentication after the grace period** — Tapping the card prompts for the PIN or password again, the same as the initial login.

No eLux configuration is required beyond enabling Imprivata as the authentication type for the device's OU — the grace period itself is controlled entirely by the Imprivata server policy below.

Configuration

Configure the grace period on the Imprivata Appliance Console:

1. Open or create a **user policy** under **User policies**.
2. Under **Authentication method options** → **Proximity card**, set the policy to **Proximity Card + PIN** or **Proximity Card + Password**.
3. Set **Grace period for second authentication factor** to the desired duration, in the format hours : minutes (0–24 hours, 0–59 minutes).



4. Save the policy and assign it to the relevant users, group, or OU.

Good to Know

- The grace-period timer resets on every successful PIN or password login — it isn't a fixed once-per-day window.
- Both the proximity card and the PIN or password modality must already be enrolled for the user before the policy can apply.
- A duration of `0` disables the grace period, even if the policy itself is enabled.
- Losing connectivity to the Imprivata agent forces fail-secure behavior — the PIN or password prompt reappears even within the grace period.

Kiosk Tap & Go in Companion Mode

Starting in eLux® 7.2609.0, an eLux endpoint can run the Imprivata virtual channel even when Imprivata is not the endpoint's own authentication type — previously, the virtual channel only came up when Imprivata was the login method. This is **companion mode**: the endpoint stays set to no authentication (`AuthType = None`), and Imprivata only manages the session inside the VDA (Virtual Delivery Agent, the Citrix® component that hosts the session) by forwarding badge taps into it.

This is different from the Tap & Go behavior described in eLux Usage, where Imprivata is the eLux device's own login method. In companion mode, the eLux login screen never shows Imprivata at all.

What the user sees

ACTION	RESULT
Tap badge at the Imprivata lock screen	Signs in to their applications inside the Citrix session.
Tap out	The VDI session locks. The eLux desktop does not show a lock screen, and the Citrix session stays connected — no reconnect happens.
Another user taps in	The VDI user switches in place. eLux does not log off or restart the session.

In companion mode, eLux doesn't log the user off when the user leaves the device, even if the Imprivata computer policy sets **Log off user and terminate SSO session**. eLux logs users off only when Imprivata is the device's own login method, see Lock or Log Off When Users Leave the Device.

Requirements

- eLux 7.2609.0 or later on the endpoint.
- The eLux endpoint set to no authentication (`AuthType = None`).

Configuration

#	SIDE	SETTING	WHERE	VALUE	REQUIRED
1	Endpoint	Authentication type	UniconOS Management Board → Device Configuration → User Authentication	`None`	Yes
2	Endpoint	`[Imprivata]`	Scout advanced	Your OneSign	Yes

#	SIDE	SETTING	WHERE	VALUE	REQUIRED
		`BootstrapUrl`	file entry on `/setup/terminal.ini`	URL, for example `https://onesign.example.com`	
3	Endpoint	`[Imprivata] AgentType`	Scout advanced file entry on `/setup/terminal.ini`	`1`, `2`, or `3`. Defaults to `3`.	Only if you need Agent Type `2`
4	VDA	`Type`	Registry key `HKLM\SOFTWARE\SSOProvider\ISXAgent`	Must match the endpoint's `AgentType`. Restart the VDA after changing it.	Yes
5	VDA	`LockRemoteSessionWithAgentOnClient`	Registry key `HKLM\SOFTWARE\SSOProvider\ISXAgent`	`1`	Yes

Desktop Chooser

Starting in eLux® 7.2611.0, Imprivata XenDesktop® automation starts the Citrix® desktop of the user who taps in. If the user has several desktops, the desktop chooser appears on the login screen, so that the user can pick the desktop to work with. The Imprivata computer policy decides whether the chooser also appears for a user with only one desktop. For the setup, see XenDesktop Automation and the Desktop Chooser.

The chooser replaces the sign-in form on the right half of the login screen. Its title is **Choose a desktop**. The list below the title shows each desktop with its icon and name. If there are more desktops than fit on the screen, the list scrolls. Below the list is the **Cancel** button.

What the User Sees

The table follows a user from the tap-in to the desktop and shows what a card tap does along the way:

ACTION	RESULT
Sign in at the login screen, for example with a card tap	**Logging in…** appears while eLux reads the user's desktops from the StoreFront™ stores. Then the chooser appears, or the only desktop starts directly, depending on the computer policy.
Click a desktop in the chooser	**Loading…** appears until the desktop is ready. Then the desktop fills the screen.
Click **Cancel** in the chooser	eLux logs the user off and shows the login screen again.
Tap the same card while the chooser is shown	The device locks. After unlocking, the chooser shows the same list again.
Tap the same card while the desktop runs	The device locks. After unlocking, the user is back in the same desktop, without the chooser. The computer policy decides whether the desktop stays connected while the device is locked.
Another user taps their card while the chooser is shown or the desktop runs	eLux logs off the first user, reads the desktops of the new user, and shows their chooser or starts their desktop.
eLux finds no desktop for the user	An error panel appears. **Retry** reads the desktops again, and **Log off** returns to the login screen.

Good to Know

- The chooser doesn't close by itself. It stays on the screen until the user picks a desktop, clicks **Cancel**, or taps a card. A user who leaves the device should tap their card to lock it.
- When a desktop session ends, for example because the user signs out of Windows, eLux notices it only when Citrix Workspace™ app has closed. After a disconnect, this takes a few seconds. After a sign-out, it can take about 20 seconds, because Windows signs the user out

before the connection closes. Users should wait until the login screen is back before the next tap, because eLux handles an earlier tap as a tap on the running session.

- On the first tap after the device starts, eLux sometimes finds no desktops and shows the error panel. In this case, the user clicks **Retry**.
- If the Imprivata policy logs users off when they leave the device, a tap of the same card logs the user off instead of locking the device. See Lock or Log Off When Users Leave the Device.

Lock or Log Off When Users Leave the Device

When users leave a shared eLux® device, they tap out with their card, press a hot key, or just walk away. Starting in eLux 7.2611.0, the Imprivata policy decides what the device does then. It either locks the device, or it logs the user off and disconnects the Citrix® session. Logging off leaves no user session on the device, so the next person at the device can't reach the previous user's desktop or applications.

You set this up on the Imprivata appliance only. Scout has no setting for it. eLux follows the hot key, the inactivity time and the inactivity warning of the policy, whether the policy locks the device or logs the user off.

What the User Sees

The computer policy setting **Inactivity lock or hot key behavior** has two options. The table shows what each way of leaving the device does with each option:

ACTION	LOCK COMPUTER AND SUSPEND SSO SESSION	LOG OFF USER AND TERMINATE SSO SESSION
Tap the card on the reader while the user is logged in (tap-out)	The device locks. The user stays logged in.	eLux logs the user off and shows the login screen.
Press the hot key of the policy	The device locks.	eLux logs the user off.
Press the eLux key combination for locking the screen, Ctrl+Alt+End by default	The device locks.	eLux logs the user off.
Press the hot key of the user policy inside the Citrix	The device locks.	eLux logs the user off.

ACTION	LOCK COMPUTER AND SUSPEND SSO SESSION	LOG OFF USER AND TERMINATE SSO SESSION
session, if the Imprivata agent runs in the session		
Leave the device without input until the inactivity time of the policy is over	The device locks.	eLux logs the user off.
Click **Lock** in the Start menu	The device locks.	The device locks. When the inactivity time is over, eLux logs the user off.
Tap the card at the lock screen	The user unlocks the device as usual.	The user unlocks the device as usual.
Another user taps their card while a user is logged in	eLux logs off the first user and logs in the new user.	eLux logs off the first user and logs in the new user.

When eLux logs the user off, it closes the user's applications, including Citrix Workspace™ app. This disconnects the Citrix session but doesn't end it. The user stays signed in to Windows on the server and can reconnect to the session later, on this device or on another one. How long a disconnected session stays available depends on the Citrix configuration. A persistent session for fast user switching keeps running, as at every log off, see Citrix Configuration.

With Imprivata XenDesktop® automation, the computer policy option **When a XenDesktop endpoint is locked** decides whether the desktop stays connected while the device is locked, see XenDesktop Automation and the Desktop Chooser.

Requirements

- eLux 7.2611.0 or later.
- Imprivata as the authentication type of the device, see Scout Settings. In companion mode, where the device itself uses no authentication, eLux doesn't log users off, see Kiosk Tap & Go in Companion Mode.

Configuration

You set up the behavior in the Imprivata Admin Console, in the computer policy of the devices:

1. Under **Computers** > **Computer policies**, open the computer policy of the devices.

2. On the **Override and Restrict** tab, in the **Challenges** section, select the **Override User Policy?** checkbox.
3. In **Hot key to lock workstation or log off**, enter the key that users press when they leave the device, for example `F4`. For the notation, see the table below.
4. Under **Inactivity lock or hot key behavior**, select what the device does when a user leaves it:
   - **Lock computer and suspend SSO session** locks the device.
   - **Log off user and terminate SSO session** logs the user off and disconnects the Citrix session.
5. On the **Walk-Away Security** tab, under **Inactivity detection** > **Keyboard and mouse**, select in **Lock workstation after** how long the device can stay without input.
6. Optional: In **Show inactivity warning**, select after how long without input a warning appears. The warning then counts down the rest of the time to the lock or the log off.
7. Save the policy.
8. If the computer policy doesn't apply to the devices yet, assign it: under **Computers** > **All Computers**, open each device, select the policy in **Apply Computer Policy**, and save.

The devices use the new settings from the next time a user logs in. Unlocking a locked device doesn't count as a login.

If **Override User Policy?** isn't selected, the hot key and the choice between locking and logging off come from the user policy of the user who logs in. In the user policy, these settings are on the **Challenges** tab: **Hot Key to Lock Workstation or Log off User** and **At Inactivity Challenge or Pressing Hot Key**, with the options **Lock Workstation/Suspend SSO Session** and **Log off User/ Terminate SSO Session**. The inactivity time and the warning always come from the computer policy.

With **Override User Policy?** selected, the computer policy decides even if its hot key field is empty. In this case, the policy has no hot key.

The hot key uses the Imprivata notation, where `\^` stands for Ctrl, `%` for Alt and `+` for Shift. Some keys, such as Delete, are written in braces. These entries are examples of the notation and the keys they stand for:

ENTRY	KEY
`F4`	F4

ENTRY	KEY
`\^l`	Ctrl+L
`%k`	Alt+K
`\^%l`	Ctrl+Alt+L
`{DEL}`	Delete

For the names of other keys, click **How do I enter a hot key?** below the hot key field of the computer policy.

eLux reserves the hot key for the whole device. While a key is the hot key, no application on the device can use it, not even in the Citrix session. So choose a key that users don't need otherwise, for example a combination with Ctrl or Alt. A hot key with Shift only, for example `+j` for Shift+J, also reacts when a user types a capital J.

If eLux can't use an entry, it ignores the hot key. The eLux key combination for locking the screen works in any case. You set it in **Keys to lock screen** of the Desktop settings.

Inactivity and Power Management

While a user is logged in, the inactivity time of the policy decides when the device locks or logs the user off. The time starts again with every input. Input at the lock screen counts too, for example a mouse movement. If **Lock workstation after** is shorter than one minute, the device locks or logs the user off after one minute.

The inactivity time replaces the screen saver time of the device's power management settings. The other power management settings work together with the policy as follows:

- With **Log off user and terminate SSO session**, the log off also replaces the **On idle** action, for example **Sleep mode**. When nobody is logged in, all power management settings apply as usual.
- With **Lock computer and suspend SSO session**, the **On idle** action, if switched on, runs together with the lock when the inactivity time is over.
- **Turn off display after** applies as set. If this time is shorter, the screen goes dark before the device locks or logs the user off. A dark screen isn't locked.

If **Lock workstation after** isn't set in the computer policy, the power management settings decide when the device locks for inactivity, as they do without Imprivata. A tap-out and the hot keys still lock the device or log the user off, as set in **Inactivity lock or hot key behavior**.

While an application keeps the screen awake, for example a video or a web meeting, the device doesn't lock for inactivity and shows no warning. With **Log off user and terminate SSO session**, eLux shows the warning and logs the user off anyway.

Good to Know

- If eLux can't read the policy at login, for example because the Imprivata appliance can't be reached, the device locks instead of logging the user off. Its power management settings then decide when it locks for inactivity.
- A policy that eLux has read at login stays in effect if the Imprivata appliance can't be reached later.
- Inside the Citrix session, the Imprivata agent uses the hot key of the user policy and passes it on to the device. Whether the device then locks or logs the user off depends on the policy that eLux has read at login. If the computer policy has the same hot key, eLux catches the key before it reaches the session. So set different keys in the two policies if the agent in the session should get its own hot key.
- Locking Windows inside the Citrix session doesn't lock the device and doesn't log the user off. If the Imprivata agent in the session is set up to pass a Windows lock on to the device, eLux handles it like the hot key.
- If a device doesn't follow a changed policy, log the user off and on again. Then check that the policy applies to the device. Also check that the device has the right **Bootstrap URL** and the certificate of the Imprivata appliance, see Scout Settings. With the bootstrap URL of another appliance, the device reads the policy of that appliance. Without the certificate, it reads no policy.

Configuration

The following steps are required to enable authentication with Imprivata. Configuration must be performed both on the Imprivata appliance and within Scout.

System Requirements

Enable ProveID Web API Access in Imprivata

Scout Settings

Citrix Configuration

VDI Access Policy

Browser SSO

System Requirements

Before beginning the configuration, verify that all required software versions are installed and prerequisites are fulfilled.

Supported Versions

- **Imprivata EAM:** 26.1
- **Scout:** 15.2507 or later
- **eLux:** 7.2509 or later

Web applications that open automatically after the Imprivata sign-in need Imprivata EAM 26.3 or later and eLux 7.2611.0 or later, see Opening web applications automatically.

Installation Prerequisites

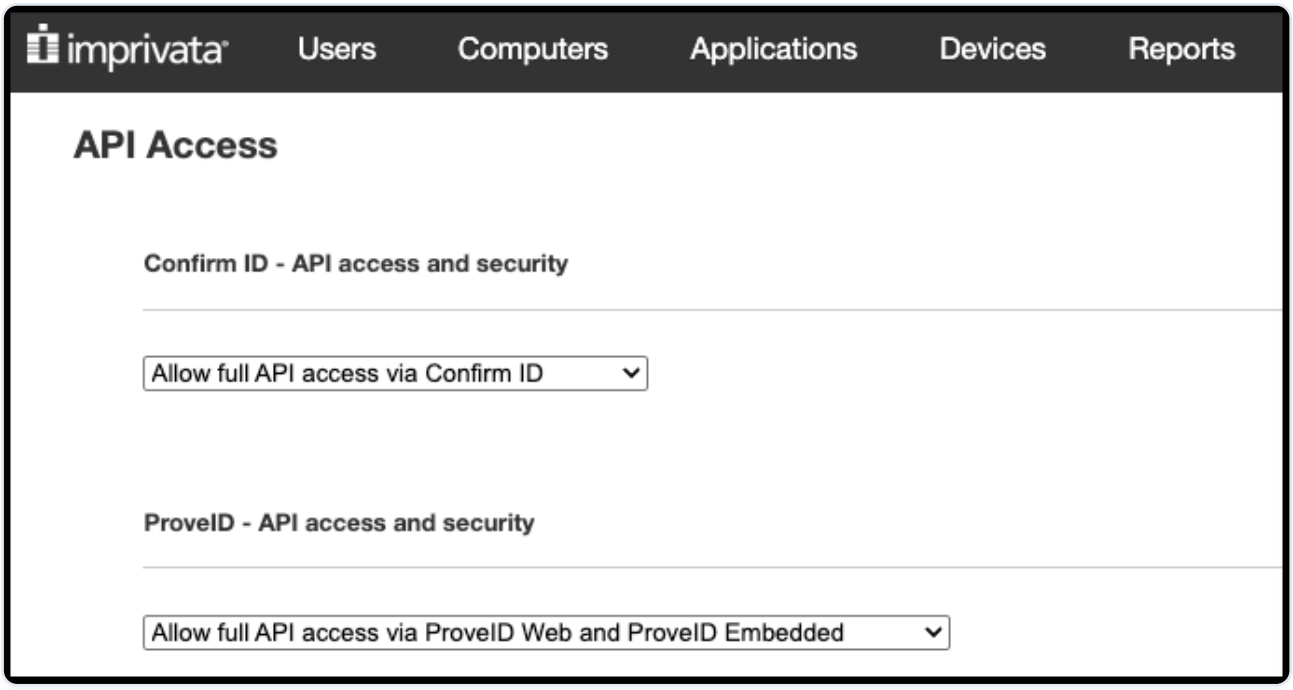
- The eLux device image must include the **Imprivata Access Management EPM package**.
- For full support of all Imprivata-related features, ensure the following optional FPM packages are also enabled:
  - Proximity Reader support
  - Fast User Switching support
  - Imprivata WebSSO runtime support, together with Google Chrome, for browser SSO and web applications at sign-in
  - Imprivata test D-Bus client/server (debugging purposes only)
- For support of pass-through logon, the **Dialog Extension** FPM of the **Citrix extensions EPM package** must be included in the eLux device image.

The Imprivata functions that are new in eLux 7.2611 are split across the **Imprivata Access Management EPM package** and the EPM packages `elux_management` and `userauth`. Take these three packages from the same eLux release. Rolling back only some of them isn't supported.

Enabling ProveID Web API Access in Imprivata

To allow eLux to integrate with Imprivata as an authentication backend, API access must first be enabled on the Imprivata appliance:

1. Log in to the **Imprivata Admin Console** with an administrator account.
2. In the upper-right corner of the console, click the **gear icon** and select **API Access**.
3. Under **ProveID – API Access and Security**, enable the option **Allow full API access via ProveID Web API and ProveID Embedded**.
4. Select the **Citrix eLux** checkbox to complete the configuration (Future 1 in older versions of the Imprivata EAM).

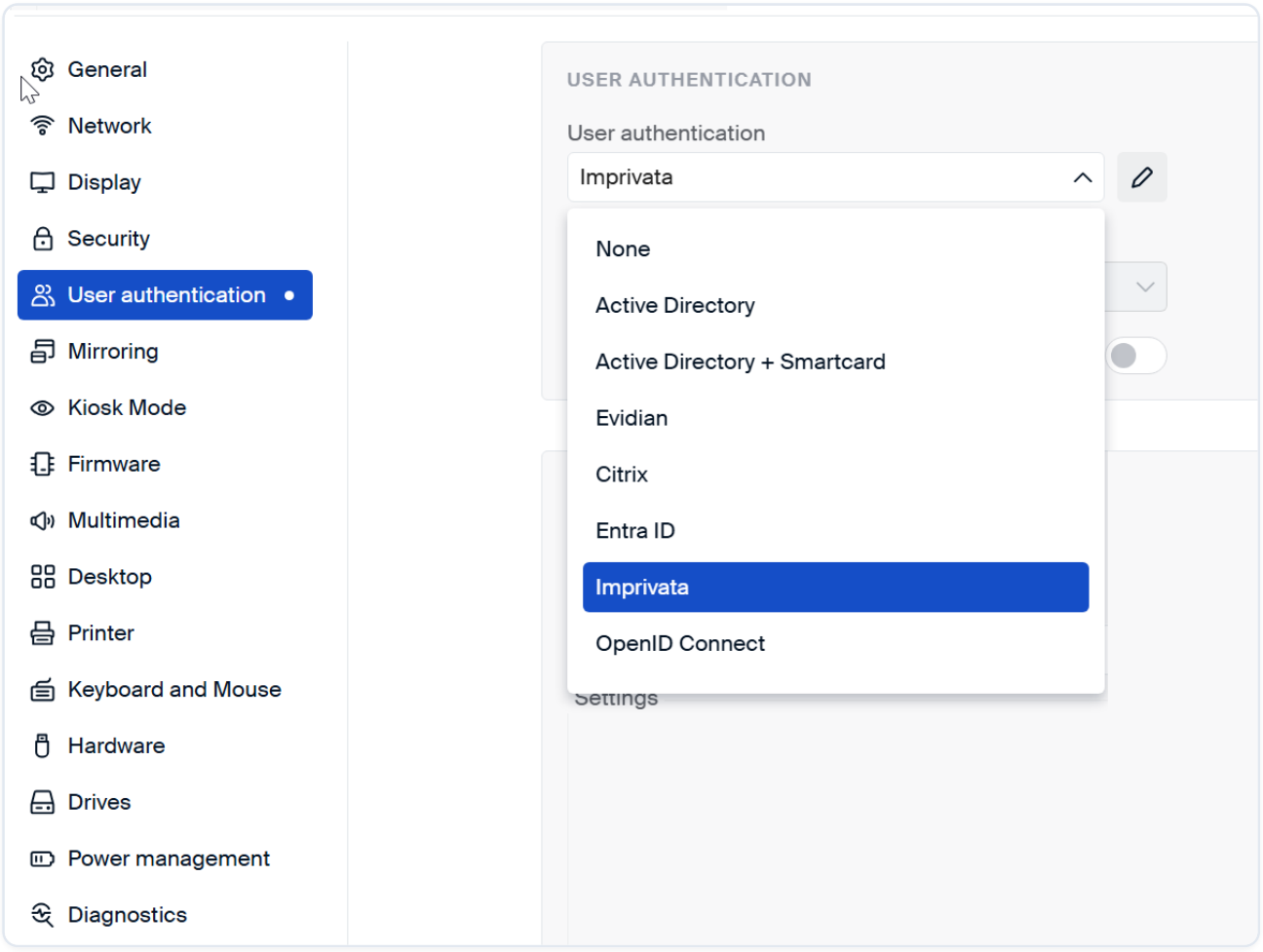


Scout

Enabling Imprivata as Authentication Type

To configure Imprivata as the authentication method, proceed as follows:

1. Open **UniconOS Management Board** and navigate to **Device Configuration**.
2. In the **User Authentication** section, click **Edit**.
3. From the drop-down menu, select **Imprivata**.
4. Click the **Edit** button below and enter the **Bootstrap URL**, which is the address of the Imprivata appliance.
5. The **timeout setting** (default: 60 seconds) defines how long the system waits for a password or PIN entry after a proximity card tap. If the entry is not completed within this period, the system automatically returns to the login screen.
6. Click **Apply/OK** to confirm the changes.



Deploying Imprivata Appliance Certificates

A certificate in **Base64 X.509 format (.crt)** must be installed on eLux devices to enable communication with the Imprivata appliance.

- By default, each Imprivata appliance is shipped with a **self-signed certificate** generated by the appliance's built-in root CA.
- This certificate is sufficient for initial testing or smaller deployments.
- However, self-signed certificates must still be signed by a **root-CA**, and the corresponding **root-CA certificate** must also be available on the client.

Many organizations prefer to replace the default certificate with one signed by a trusted authority. This can be:

- An **internal enterprise CA** (e.g., Microsoft Active Directory Domain Enterprise Root CA), or
- A **publicly trusted CA** such as DigiCert.

In such cases, the **complete certificate chain** (root CA, intermediate/subordinate CA) must be exported in Base64 X.509 format and deployed to all eLux devices via Scout.

OPTION 1: USING THE IMPRIVATA SELF-SIGNED ROOT CERTIFICATE

1. Log in to the **Imprivata Appliance Console** and open the **Security** tab.
2. If the following message is displayed: “The SSL certificate for this appliance has been self-signed by the certification authority (CA) on this appliance. Download the certificate of this CA.”*
3. Click **Download the certificate**.
4. The file will be saved as a **.crt file**.

This root certificate must then be deployed to your eLux devices via **Scout** to establish a trusted connection between clients and the appliance.

☞ For guidance on how to deploy certificates via Scout, please refer to Certificates.

OPTION 2: USING A TRUSTED THIRD-PARTY OR ENTERPRISE CA

If the Imprivata appliance has been configured with a certificate signed by a corporate or public CA, the root and intermediate certificates cannot be downloaded from the appliance directly.

In this scenario:

1. Use a supported browser (e.g., **Microsoft Edge** or **Google Chrome**) to access the **Imprivata Appliance Console**.
2. Export the complete certificate chain (root CA, intermediates, and appliance certificate).
3. Ensure the chain is saved in **Base64-encoded X.509 format (.crt)**.
4. Deploy the certificate chain to your eLux devices via **Scout**.

> ***Note:*** Please make sure that the time and date settings on your eLux devices are accurate. Incorrect time settings can lead to certificate validation failures, preventing successful communication with the Imprivata appliance.

Citrix Configuration

The configuration of Citrix VDI sessions is managed through **Scout**.

For a detailed walkthrough, please refer to Connecting to a Citrix® farm.

When configuring Citrix sessions in the Imprivata context, it is important to distinguish between **private user sessions** and **persistent sessions**:

Private User Session

A private Citrix virtual desktop or application session provides each user with access to their own set of applications and desktops.

- The session is terminated upon logoff.
- Best suited for **classic, user-centric workloads** such as office and communication applications.

CONFIGURATION OF PRIVATE USER SESSIONS

In order to configure private user sessions in Scout, proceed as usual but ensure the following settings are applied in the configuration of your Storefront application:

- In advanced settings:
  - Timed logoff: **Logoff after** is not set (disabled).
  - Manual logoff: **Logoff server and disconnect sessions** is enabled.

Persistent Session

For workflows requiring **Fast User Switching (FUS)** and immediate access to shared clinical applications (e.g., Electronic Health Records), persistent Citrix session is required.

- These sessions remain active even after a user logs off.
- The application is instantly available to the next user logging in.
- The Imprivata Agent running inside this Citrix session is informed of changes in regard to the user session (i.e. logon/logoff/locking/unlocking).
- Each eLux client configured for a persistent session requires a **dedicated service account**.

CONFIGURATION OF A PERSISTENT SESSION

In order to configure the persistent session in Scout, proceed as usual but ensure the following settings are applied in the configuration of your Storefront application:

- Username and password fields are set to the dedicated service account credentials for this eLux client (see tip below).
- In advanced settings:
  - Timed logoff: **Logoff after** is not set (disabled).
  - Application Reconnection is set to **Do not reconnect**.
  - Manual logoff: is set to **Logoff only server**.
- In *Free parameters* add the following entries:
  - *Variable*: Persistent; *Value*: true
  - *Variable*: Hidden; *Value*: true

TIP:

The %HOSTNAME% macro can be used in the username field to automatically append the eLux client hostname to the service account name. This allows the use of a single service account prefix for all eLux clients (e.g., "citrixsvc_%HOSTNAME%").

Note:

When using version 2508 of the Citrix CWA EPM package, please also add the following entry under advanced file entries in Scout: *file*: /setup/sessions.ini *Section*: ICADefaults *Key*: StorebrowseIPCDisabled *Value*: true

> **Important:**

> When using persistent and private sessions in parallel, please ensure that different Storefront URIs are used to avoid conflicts.

VDI Access Policy

Imprivata VDI access policies and Scout Citrix configuration aren't mutually exclusive. For the areas Imprivata covers, it acts as the **primary** source and **Scout is the fallback**, so the same resource doesn't have to be configured twice.

Today this applies to VDI access policies and, from eLux® 7.2611.0, to the hot key and the inactivity settings of the computer policy, see Lock or Log Off When Users Leave the Device. Other Imprivata computer-policy areas, such as Secure Walk Away and the desktop behavior, aren't combined with Scout this way yet.

A VDI access policy is configured on the Imprivata server: under `Computers > Virtual Desktops`, and on the **Workspace Access** tab of the computer policies and the user policies. It defines which Citrix desktops and published applications a user is entitled to, and whether they launch automatically. When the user signs in at the eLux device through Imprivata, the device reads their VDI access policy and starts those Citrix resources automatically with single sign-on, using the credentials already presented at Imprivata login — the user doesn't re-enter anything.

This is where Imprivata and Scout meet: Citrix Configuration distinguishes between a **private** session (launched for the individual signed-in user) and a **persistent** session (a shared session under a common account). A VDI access policy changes how the **private** session behaves; the **persistent** session is left to Scout.

Configuring on the Imprivata Server

A VDI access policy is built from three layers. All three must align for a Citrix resource to auto-launch — if any layer omits it, the resource doesn't start.

In older Imprivata versions, the **Workspace Access** tab is called **Virtual Desktops**.

1. **`Computers > Virtual Desktops`** (inventory) — Defines the pool of Citrix stores and their published resources.
    - **XenDesktop:** Add each StoreFront server URL and enable **Allow authentication from XenDesktop-enabled devices**.

- **XenApp:** Add each StoreFront server URL and list the exact published application/delivery-group name(s); enable **Allow authentication from XenApp-enabled devices**.

2. `Computers > Computer policies`, **Workspace Access** tab (per device) — Chooses how a computer uses the inventory.
   - **XenDesktop:** Enable **Automate access to Citrix XenDesktop**; choose the desktop-launch prompt behavior; set the **When a XenDesktop endpoint is locked** option; select the available Citrix servers. For what these options do, see XenDesktop Automation and the Desktop Chooser.
   - **XenApp:** Enable **Automate access to Citrix XenApp** and **Enable Published Applications**; set the **When the device is locked** option; select the available Citrix servers.
   - Assign the computer policy to the eLux device: under **Computers** > **All Computers**, open the device, select the policy in **Apply Computer Policy**, and save.
3. `Users > User policies`, **Workspace Access** tab, **Virtual Desktop** section (per user) — Defines what runs for the signed-in user; this is the master switch for VDI access automation.
   - **Enable virtual desktop access automation** — if disabled, no resources are injected and the Scout fallback applies.
   - **XenDesktop:** Enable **Automate access to full VDI desktops** and select **Citrix** as the vendor. Starting in eLux 7.2611.0, the device reads the user's desktops from the stores in the computer policy and ignores the list **Then, on that desktop, launch the following applications**. Leave this list empty.
   - **XenApp:** Enable **Automate access to applications or published desktops**, choose the roaming option, and select which applications auto-start.
   - Under **Authentication > Primary factors**, select the login factors (for example Password and Proximity Card).
   - Assign the user policy to the user.

Configuring on the Scout Side

- **Private session** — No Scout changes are needed; the resource is left to the Imprivata VDI access policy.
- **Persistent session** (used with fast user switching — an always-on shared application, for example an EHR app) — Configure it in Scout as a persistent StoreFront application as usual.

It keeps running across user switches, and the Imprivata resource for that store is dropped instead.

Precedence Between Imprivata and Scout

Starting in eLux 7.2611.0, the precedence rules in this section apply to XenApp automation only. XenDesktop automation reads the desktops from the stores instead, see XenDesktop Automation and the Desktop Chooser.

Precedence is decided per Citrix resource, matched on the same store (StoreFront URL) and the same published resource or delivery group:

SITUATION	WINNER	RESULT
The user's Imprivata VDI access policy matches a Scout **private** session	Imprivata	The Imprivata resource starts with single sign-on. The matching Scout private session is suppressed while the user is signed in and restored automatically at logoff or user switch — it's never deleted.
The user has no applicable Imprivata VDI access policy	Scout (fallback)	Nothing is injected from Imprivata; the Scout-configured private session runs as configured.
A Scout **persistent** session exists on the store	Scout	The Imprivata resource for that store is dropped; the persistent session keeps running untouched, including across a fast user switch.

Scout wins on persistent sessions because Citrix Workspace App can't open one store under two identities at once, and an Imprivata VDI access policy can't express a shared, common-user login — it always launches as the current signed-in user. A persistent session must therefore stay a Scout configuration.

XenDesktop Automation and the Desktop Chooser

Starting in eLux 7.2611.0, a shared device can start the Citrix desktop of the user who taps in, instead of a desktop that Scout assigns to the device. With XenDesktop automation, eLux reads

the desktops that Citrix publishes to the user from the StoreFront stores in the Imprivata computer policy. A user with several desktops picks one in the desktop chooser. For what the user sees on the device, see Desktop Chooser.

eLux reads the Imprivata policy and the user's desktops again at every login, for example when a user taps in at the login screen. Users with different desktops can therefore share the same device. Unlocking the device doesn't count as a login. When you change a user's desktops in Citrix, or the Imprivata policy, the change applies at the user's next login. You don't have to change the device configuration in Scout.

REQUIREMENTS

- eLux 7.2611.0 or later, with Citrix Workspace app for Linux 2608 or later.
- Imprivata as the authentication type, see Scout Settings.
- Boot-to-VDI, see Boot-to-VDI. XenDesktop automation works only on boot-to-VDI devices. On a device without boot-to-VDI, eLux reads the stores but doesn't start a desktop, and the user gets the eLux desktop without a Citrix session. Restart the devices after they get the boot-to-VDI configuration.
- A user policy with **Automate access to full VDI desktops**. Users whose user policy selects **Automate access to applications or published desktops** get XenApp automation instead, without the chooser. XenApp automation needs a device without boot-to-VDI, so a device serves either users with XenDesktop automation or users with XenApp automation.
- No Persistent Home for ICA®: `PersistentHomeApplications` contains neither `ica` nor `ALL`, see Persistent User Configurations. Persistent Home for ICA would keep the StoreFront credentials of one user on the device for the next user.
- No prompt for credentials: eLux signs in to the stores with the user's Imprivata credentials. XenDesktop automation that prompts the user for credentials isn't supported with the desktop chooser.

COMPUTER POLICY OPTIONS

The options for XenDesktop automation are in the Imprivata computer policy, on the **Workspace Access** tab, in the **Citrix** section under **Citrix XenDesktop**. In the XenApp part of the same section, leave **Automate access to Citrix XenApp** off. A boot-to-VDI device doesn't start XenApp resources.

To turn XenDesktop automation off, clear **Automate access to Citrix XenDesktop**, or move the users to a user policy without XenDesktop automation. The device configuration in Scout has no setting for it.

The drop-down list below **Automate access to Citrix XenDesktop** decides when eLux shows the chooser. What the user gets also depends on how many desktops eLux finds:

DESKTOPS FOUND	PROMPT THE USER ONLY IF THEY HAVE MULTIPLE DESKTOPS	ALWAYS PROMPT THE USER TO CHOOSE THEIR DESKTOP
None	An error panel appears.	An error panel appears.
One	The desktop starts directly.	The chooser shows the one desktop.
Two or more	The chooser shows all desktops.	The chooser shows all desktops.

eLux reads every store that is selected under **Available Citrix servers** and shows the desktops of all these stores in one list. The list contains desktops only, not published applications, so a user who has only published applications gets the error panel. The list shows the name of each desktop but not its store, so give desktops on different stores different names.

> **Note:**
>
> eLux signs in to the selected stores with the user's credentials. Select only stores that you trust, and limit who can change the computer policy.

> **When a XenDesktop endpoint is locked** decides what happens to the running desktop when the user locks the device, for example with a tap of their card:

OPTION	RESULT
Keep the XenDesktop client and user session active	The desktop stays connected behind the lock screen. After unlocking, the user continues in the same session.
Shutdown the XenDesktop client and disconnect the user session	eLux disconnects the desktop a few seconds after the lock. After unlocking, eLux reconnects the same desktop, without the chooser.

This option applies only when the device locks. If **Inactivity lock or hot key behavior** is set to **Log off user and terminate SSO session**, a card tap logs the user off and disconnects the desktop instead, see Lock or Log Off When Users Leave the Device.

CITRIX SETTINGS IN SCOUT

With XenDesktop automation, the Imprivata policy decides which desktop starts. Citrix settings that remain in the device configuration in Scout affect this as follows:

SCOUT CONFIGURATION	RESULT
Boot-to-VDI resource to launch (`ScenarioResource`)	eLux starts this resource for every user. It doesn't read the stores, and the chooser doesn't appear. Leave `ScenarioResource` empty for XenDesktop automation.
Private sessions and other StoreFront applications	eLux doesn't start them, because a boot-to-VDI device starts only its boot-to-VDI session.
Persistent StoreFront application for fast user switching	Not supported. Fast user switching can't be combined with XenDesktop automation.

Imprivata Failover & Recovery

Goal

Ensure uninterrupted authentication by automatically:

- Switching to a **failover site** when all configured home sites are unavailable.
- Periodically checking whether a home site has recovered.
- Switching back to a working home site without user interaction.
- Recovering automatically from offline mode when any site becomes reachable.

Prerequisites

Before you begin, ensure the following:

- eLux **7.2601.0** or later is installed.
- Imprivata is configured and operational.
- A valid **bootstrap URL** is defined.
- Network access to home and failover sites is properly configured.
- Administrative access to run network test commands (for validation scenarios).

How Failover Recovery Works

NORMAL OPERATION

- The device connects to a configured **home site**.
- Authentication works as expected.

HOME SITE UNAVAILABLE

- If the home site becomes unreachable, the device automatically switches to a configured **failover site**.
- Users can continue logging in without interruption.
- While connected to the failover site, the device periodically checks whether a home site is available again.

AUTOMATIC SWITCH BACK

- Every `failoverSiteTimeoutSeconds` (default: 5 minutes), the device attempts to contact the home sites.
- As soon as a home site responds, the device automatically switches back.
- No user interaction is required.

OFFLINE MODE

If both home and failover sites are unreachable:

- The device enters **offline mode**.
- Authentication is not possible.
- Every `offlineTimeoutSeconds` (default: 5 minutes), the device retries all configured sites.
- As soon as any site becomes reachable, the system reconnects automatically.

Configuration

Failover behavior can be configured either locally or centrally.

- Local file: `/setup/terminal.ini`
- Central management: Configurable via device profile in Scout

Failover behavior is controlled under the **"Imprivata"** section.

PARAMETER	DESCRIPTION	DEFAULT
`failoverSiteTimeoutSeconds`	How often the system checks for a recovered home site while connected to a failover site.	**300 s** (5 min)
`offlineTimeoutSeconds`	How often the system retries connecting when all sites are unreachable.	**300 s** (5 min)

> **Tip:**
>
> Lower values mean faster recovery but slightly more network traffic. The default of 5 minutes is a good balance for most environments.

Browser SSO

Browser SSO extends Imprivata single sign-on into web applications opened in Google Chrome. After a user authenticates through Imprivata, Chrome signs them into supported web applications (SAML or OIDC) automatically, without asking for a password again.

Starting in eLux® 7.2611.0, Chrome can also open each user's web applications by itself right after the Imprivata sign-in, see Opening web applications automatically.

> **Note:**
>
> Browser SSO is supported in Google Chrome only. It doesn't work in Chromium.

Prerequisites

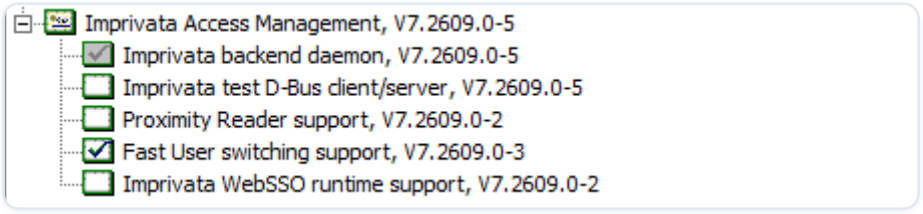
- The eLux device is configured for Imprivata authentication.
- On the Imprivata appliance, the user policy allows single sign-on: on its **Single Sign-On** tab, **Allow users single sign-on access to applications** is selected. Imprivata Cloud Connect is used to create an OpenID Connect (OIDC) application. Both are covered step-by-step in Imprivata's own documentation: OneSign SSO (OIDC).

- The device can reach the Chrome Web Store (`clients2.google.com`), because Chrome installs the Imprivata browser extension from there.

Enabling Browser SSO in the IDF

1. In the **Navigation** menu of UniconOS Management Board, click **Software**, and then click **Images** in the **Software** menu.
2. Open the image of the devices that you want to change.

   For how to change an image, see Changing an image.
3. In **Defined packages**, click **Imprivata Access Management** to show its features.
4. Tick **Imprivata WebSSO runtime support**.
5. Click **Google Chrome Web browser** to show its features.
6. Tick the **Google Chrome** option, if it isn't already enabled.
7. Click **Solve dependencies**, and then click **Save image**.
8. In Scout, assign the updated image to the device or OU and start the update.



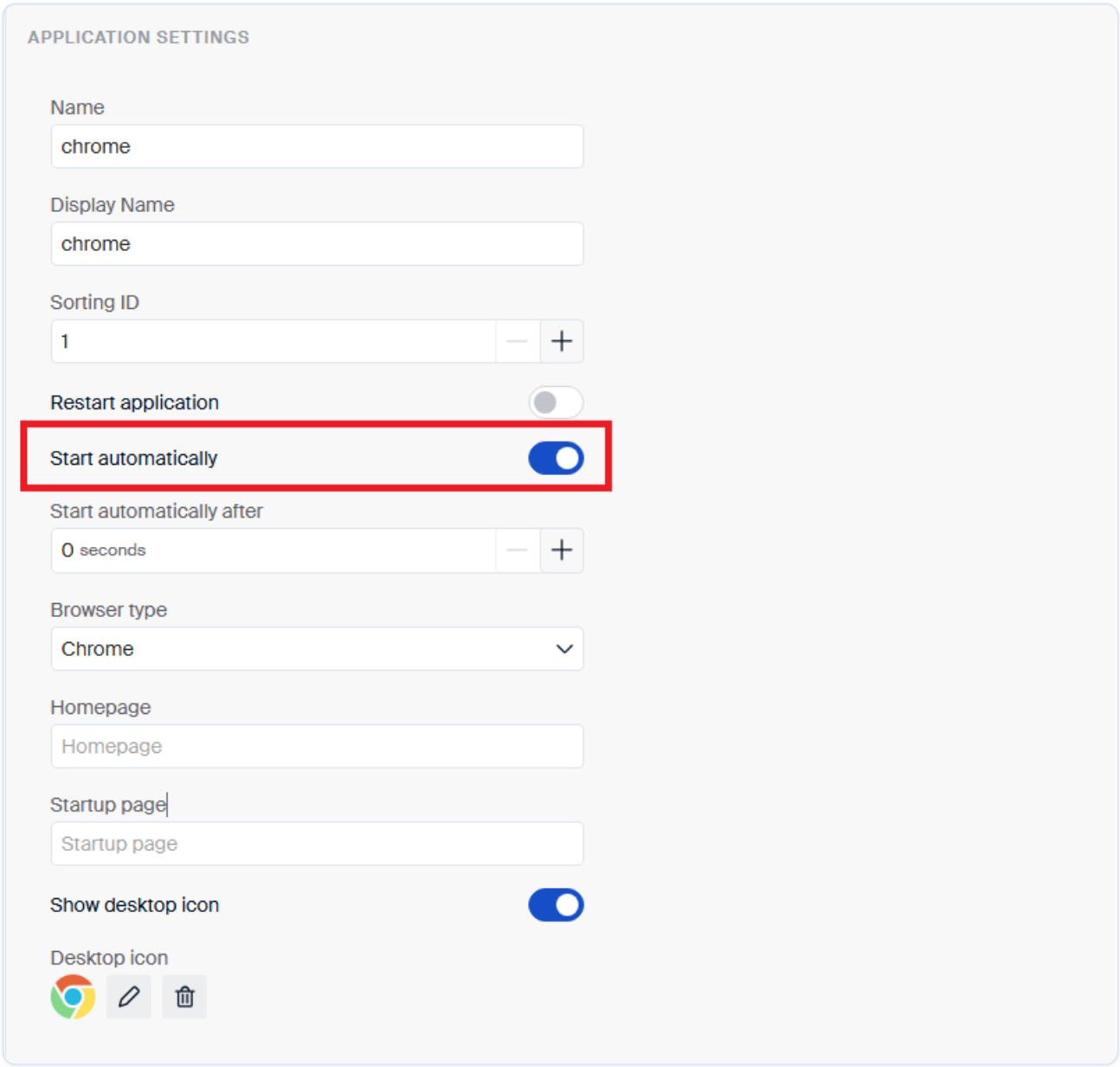
> **Important:**

> Imprivata WebSSO runtime support depends on Google Chrome. If Google Chrome isn't also enabled in the same IDF, the update fails.

Starting Chrome automatically (optional)

This step is optional — a user can also open Chrome themselves. To have Chrome open on its own right after login, enable **Start automatically** in the Chrome application settings in Scout (**Browser type**: Chrome):

1. In Scout, open the application settings for the Chrome application.
2. Turn on **Start automatically**.
3. Click **Apply/OK** to confirm the changes.



To open each user's own web applications instead, use the Imprivata policies, see Opening web applications automatically.

Verifying Browser SSO

1. Log in on the device with Imprivata.
2. Open Chrome (automatically, if you enabled **Start automatically**, or launch it yourself). The Imprivata browser extension is already installed — the user never installs it.
3. Open the target web application.

The web application signs the user in automatically, without asking for a password.

Opening web applications automatically

Chrome can open a user's web applications right after the Imprivata sign-in, and the Imprivata browser extension signs the user in to them. So the user can start working without opening Chrome or typing an address. Each user gets their own web applications, also on a device that several users share.

You choose the web applications in the Imprivata Admin Console, in a computer policy and a user policy. There are no settings for this in Scout. Web applications open only if both policies allow it.

REQUIREMENTS

- Imprivata EAM 26.3 or later. In older versions, the Imprivata Admin Console doesn't have the settings for web applications. Users sign in as before, but no web applications open.
- eLux 7.2611.0 or later.
- Browser SSO is set up, see Prerequisites and Enabling Browser SSO in the IDF. Without **Imprivata WebSSO runtime support** in the image, eLux doesn't read the policies for web applications.

SETTING UP THE COMPUTER POLICY

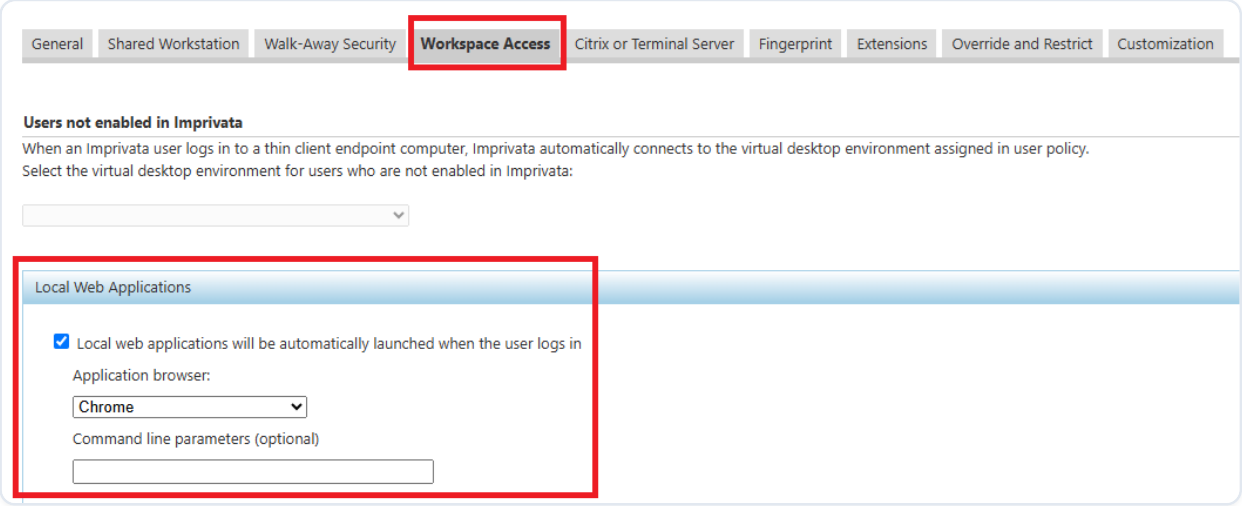
The computer policy allows web applications to open automatically on the devices that it applies to. It also decides which browser opens them, and with which Chrome options.

1. In the Imprivata Admin Console, click **Computers** > **Computer policies**.
2. Click the computer policy for the devices.
3. Click the **Workspace Access** tab.
4. Under **Local Web Applications**, select **Local web applications will be automatically launched when the user logs in**.
5. In **Application browser**, select **Default** or **Chrome**. On eLux, both open Google Chrome.

   If you select another browser, for example **Edge**, eLux opens no web applications at all, not even in Chrome.
6. If Chrome should start with particular options, enter its switches in **Command line parameters (optional)**, for example `--start-fullscreen` for a full-screen window. Separate several switches with spaces. A switch itself can't contain a space.

   eLux passes only switches to Chrome, that is, words that begin with a hyphen. It drops any other word, for example a URL: enter URLs in the user policy instead. It also drops switches that would turn off the Imprivata browser extension or weaken the security of the browser, for

example `--incognito`, `--guest`, `--disable-extensions`, `--remote-debugging-port`, `--user-data-dir` or `--no-sandbox`.

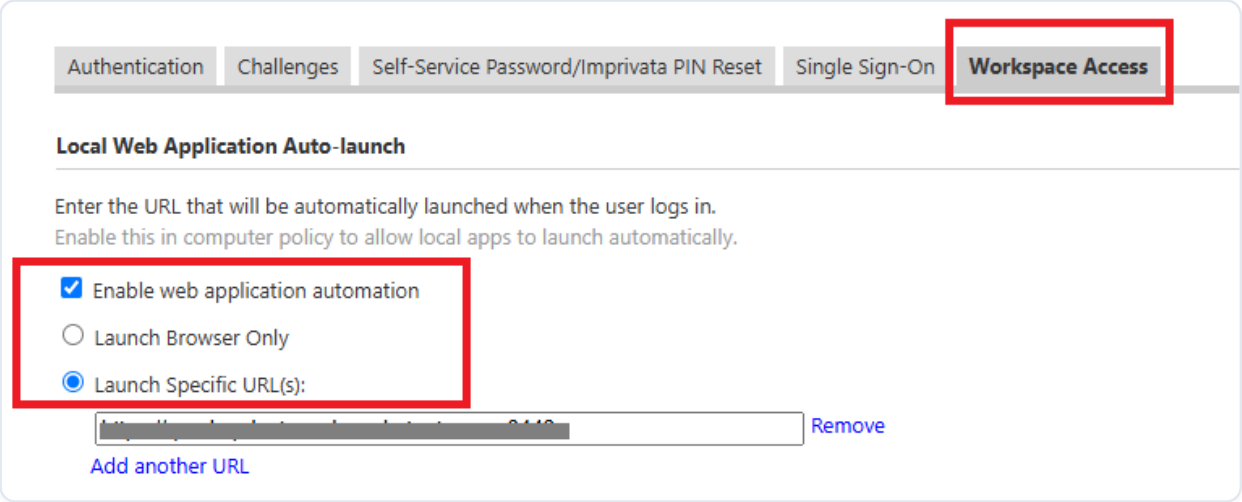


7. Save the policy.
8. If the computer policy doesn't apply to the devices yet, assign it: under **Computers** > **All Computers**, open each device, select the policy in **Apply Computer Policy**, and save.

SETTING UP THE USER POLICY

The user policy decides for each user whether web applications open, and which ones.

1. In the Imprivata Admin Console, click **Users** > **User policies**.
2. Click the user policy for the users.
3. Click the **Workspace Access** tab.
4. Under **Local Web Application Auto-launch**, select **Enable web application automation**.
5. Select what Chrome opens:
   - **Launch Specific URL(s)**: Chrome opens the URLs that you enter, each in its own tab of one window, in the order of the list. To add more URLs, click **Add another URL**.
   - **Launch Browser Only**: Chrome opens on its start page, without a web application.



6. Save the policy.
7. Make sure that the user policy applies to the users who should get web applications.

WHAT HAPPENS ON THE DEVICE

The table follows a user from the sign-in to the logoff and shows what eLux does with the web applications along the way:

SITUATION	WHAT ELUX DOES
The user signs in with Imprivata, for example with a card tap	The desktop appears first. Then Chrome opens the web applications, and the Imprivata browser extension signs the user in to them. This can take up to about half a minute. At the first sign-in after the device starts, Chrome first installs the extension, which can take up to about a minute.
The user locks the device, for example with another tap of the same card	Chrome keeps running behind the lock screen. After unlocking, the user continues in the same window.
The user closes Chrome	Chrome stays closed, also after a lock and unlock. It opens again at the user's next sign-in.
Another user taps their card	eLux closes the previous user's Chrome with all its windows, even if the Imprivata policy is set to keep applications open on a user switch. Then it opens the web applications of the new user. If the policies of the new user don't open web applications, no browser opens.

SITUATION	WHAT ELUX DOES
The user logs off	eLux closes Chrome with all its windows.

GOOD TO KNOW

- eLux reads both policies at every sign-in. If you change a policy, the change applies at the user's next sign-in, not in a running session.
- With `--start-fullscreen`, Chrome doesn't show its tabs. Users switch between the web applications with Ctrl+Tab.
- The desktop shows an icon for the web applications. It's named after the host of the first URL, for example **intranet.example.com (+1)** for two URLs. With **Launch Browser Only**, it's named **Web Application**.
- A Chrome application from Scout that starts automatically in the same session opens in the same Chrome browser as the web applications. Its page stays open, but the command line parameters of the computer policy, for example `--start-fullscreen`, then may not apply. The same happens if the user opens Chrome while eLux prepares the web applications. To make sure that the command line parameters apply, turn off **Start automatically** for the Chrome application in Scout.

TROUBLESHOOTING

If the web applications don't open as expected, check the possible causes for what the user sees:

WHAT THE USER SEES	POSSIBLE CAUSE
No web applications open after the sign-in	The computer policy or the user policy doesn't allow web applications, or it doesn't apply to this device or user.
No web applications open after the sign-in	The image doesn't contain **Imprivata WebSSO runtime support** or Google Chrome.
No web applications open after the sign-in	The Imprivata appliance is older than EAM 26.3.
The web applications open, but they ask the user to sign in	The Imprivata browser extension wasn't ready within 60 seconds after the sign-in. Check that the device can reach the Chrome Web Store, see Prerequisites.

If no web applications open, the device logs the reason in `/var/log/unicon/ucimprivatad.log` or `/var/log/unicon/ucsettingsd.log`. You find both files in the diagnostic files of the device.

License Activation Service

License Activation Service (LAS) is the next generation technology for product activations across the suite of Citrix products.

The Citrix License Server integration has been released with the Scout version 15 2508 LTSR.

From UniconOS Management 2611, Scout needs Citrix License Server 2607 (build 56200) or later. You enter the License Server when you install Scout, and you can change it later in UniconOS Management Board, see Licensing.

To see more information about the Citrix License Server and the licensing process, please follow the following links:

- Citrix License Server
- Citrix License Manager

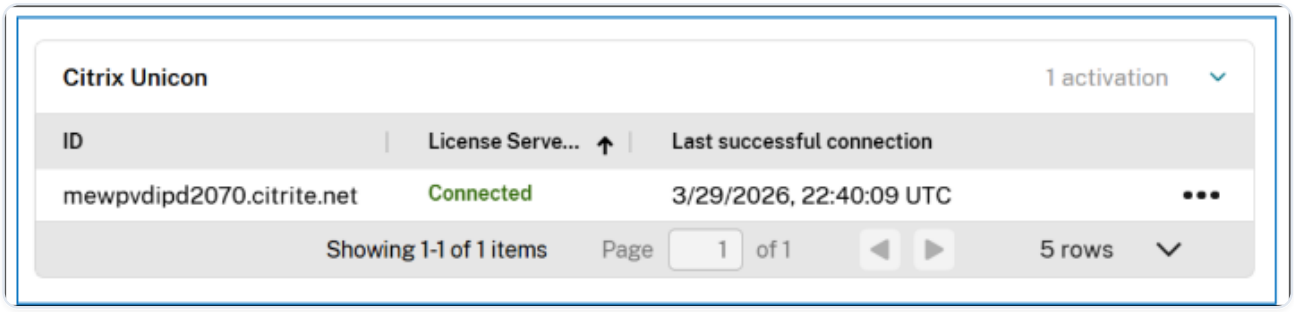
> **Note:**

> File based licensing system (including perpetual eLux and Scout licences and the Enterprise Subscription operating mode), traditionally used for activating various on-premises components, will be End of Life (EOL) on April 15, 2026. License Activation Service (LAS) is the next generation technology for product activations across the suite of Citrix products. LAS will be the only way to activate and license Citrix on-premises components after April 15, 2026. To remain supported, your Scout deployment must be on a LAS compatible version. The minimum required Scout version that is LAS compatible is Scout 15 2508 LTSR. Should you have questions or concerns, contact Customer Care. Citrix might limit or suspend your Citrix Maintenance for non-compliance with these requirements without liability in addition to any other remedies Citrix might have at law or equity. These requirements don't apply where prohibited by law or regulation.

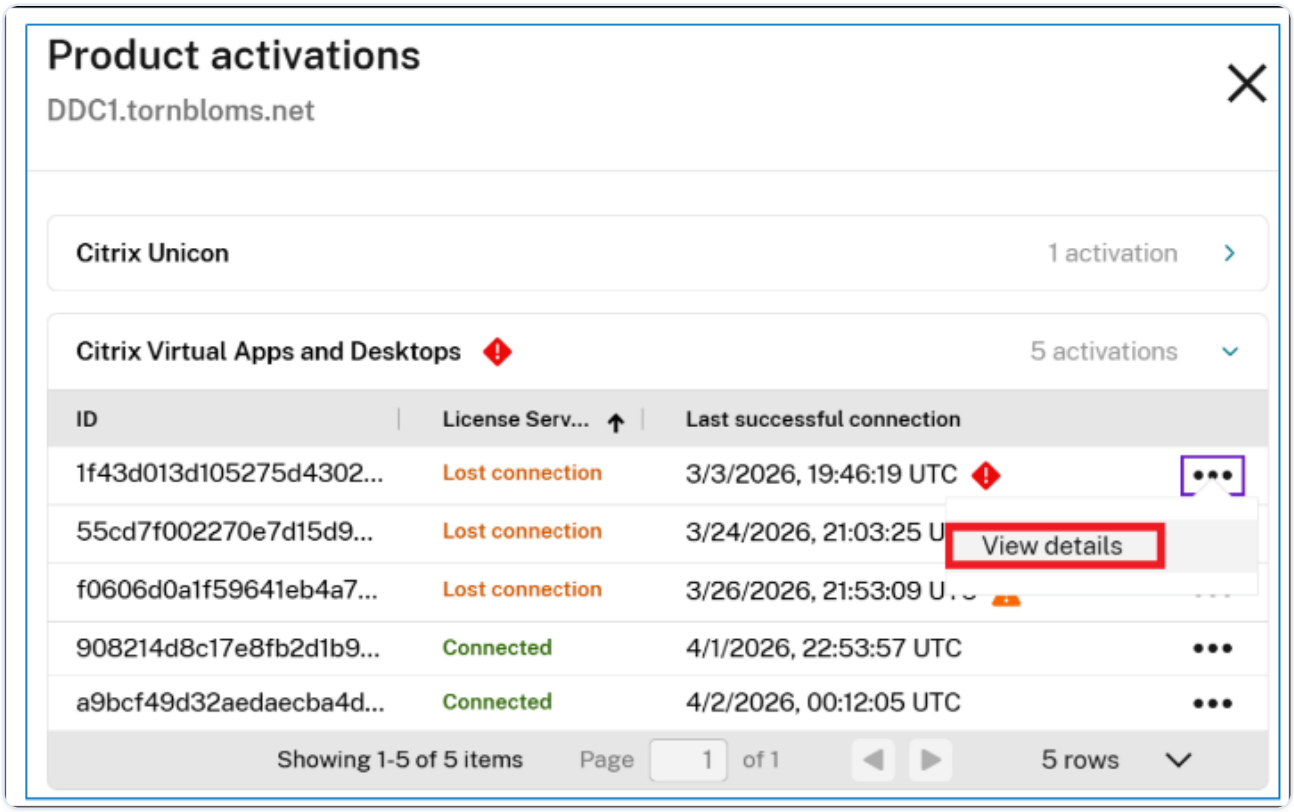
Troubleshooting

In the Citrix Cloud, navigate to **Licensing -> licensed deployments -> license Servers**

Click the expand button **>** across Citrix Unicon to see the list of Unicon Scout that were activated.



Click **View details** across the activations to see the granular details.



DeploymentId: The deployment ID

InstanceId: The Scout Server instance id

Hostname: The Scout Server host name

Scout Server Telemetry information

- Scout Server Telemetry data

Scout Server telemetry data elements

Headers

KEY	DESCRIPTION	EXAMPLE
`ver`	The version of the telemetry event.	`1.0`
`id`	A unique identifier of the event.	`0924268E-F1B9-4FCA-942E-45E63DE050ED`
`type`	The event type used to identify feature usage events.	`Scout.Device.Count`
`prod`	Identify the originating product and deployment of the usage event.	`UniconScoutServer`
`prodVer`	Scout Server product version.	`25.08.0`
`st`	The timestamp when the event is generated by the Scout Server.	`2025-08-16T09:05:00Z`

Payload

FIELD	DESCRIPTION	EXAMPLE
`scoutServer`	The ID of the server sending the event.	`3E693C53-B290-4765-9B24-88F927E91459`
`deviceCount`	Count of devices in the Scout database	`42`
`serverCount`	Count of Scout Servers in the Scout database	`1`
`timestamp`	ISO formatted time of event	`2025-08-16T09:05:00Z`

Moving the AVD client to the new Entra application

The Azure Virtual Desktop (AVD) client on UniconOS devices signs users in with Microsoft Entra ID. To do this, it identifies itself with the ID of an application that is registered in Entra ID. If users also sign in to UniconOS itself with Entra ID, this sign-in uses the same application by default, so that the AVD client can reuse it.

With UniconOS 7.2611 and UniconOS 7.2603.3000 (2603 LTSR CU3), the AVD client uses a new application, with a different application ID. In your tenant, it is a separate enterprise application, and nothing carries over from the old one. Admin consent, Conditional Access policies, user assignments and allowlists that you set up for the old application don't apply to the new one.

An Azure Virtual Desktop or Entra ID setup that works today can therefore stop working after the update. Sign-ins can fail, and security settings such as Conditional Access policies can stop applying without any warning. Before you update the devices, prepare your tenant, see Preparing your tenant.

The two applications have these application IDs:

APPLICATION	APPLICATION ID
New, used by default	`491bb6b0-7940-4e0f-bb52-b2989db8680b`
Old, will be retired	`65a544d1-2825-432e-9083-d73f9c786d92`

> **Important:**
>
> When the old application is retired, every device that still uses it fails to sign in to Azure Virtual Desktop. This includes devices that work today and that nobody has changed, so doing nothing isn't an option. Prepare your tenant and update your devices before the old application is retired.

Preparing your tenant

You need administrator rights in your tenant for the changes in Entra ID. Make them before your devices start to use the new application.

The steps in this section cover the settings that are affected most often. Some of them may not apply to you. Your tenant can also refer to the old application in other places, for example in custom policies, scripts, third-party tools, SIEM rules or automation.

Granting admin consent

Admin consent grants the new application its permissions for all users of your tenant. The application requests these permissions:

PERMISSION	PURPOSE
`openid`, `profile`, `email`	Signing the user in
`offline_access`	A refresh token, which is required for single sign-on to Azure Virtual Desktop
`https://www.wvd.microsoft.com/User.Access`	Listing and starting the user's Azure Virtual Desktop resources

Admin consent works only if the Azure Virtual Desktop service principal exists in your tenant. Tenants that already use Azure Virtual Desktop have it.

1. In a browser, open the admin consent URL of the new application. Replace `{tenant-id}` with the ID of your tenant:

   `https://login.microsoftonline.com/{tenant-id}/adminconsent?client_id=491bb6b0-7940-4e0f-bb52-b2989db8680b`

2. Sign in with an account that can grant tenant-wide admin consent, for example one with the Cloud Application Administrator role.
3. Review the requested permissions, and accept them.

The new application now appears among the enterprise applications of your tenant.

Moving settings to the new application

Next, give the new application the settings that the old one has. Find the two enterprise applications by their application ID, not by their name.

1. Add the new application to every Conditional Access policy that refers to the old one.

   Take particular care with this step. A policy that refers only to the old application doesn't apply to the new one, and nothing tells you so: sign-ins succeed, nobody reports a problem, and the control is gone.

2. If the old application has **Assignment required?** set to **Yes**, set it to **Yes** for the new application too, and assign the same users and groups. If you skip this, the new application doesn't limit who can sign in, and you get no warning.
3. If you allow applications through allowlists, for example by application ID or by publisher, add the new application. Otherwise, the allowlist blocks it.
4. Change sign-in log filters and access reviews that use the old application ID to the new one. Otherwise, your reports miss the sign-ins of updated devices.
5. Check other settings that use the application ID, for example sign-in risk policies, session controls such as sign-in frequency or app-enforced restrictions, continuous access evaluation, Global Secure Access app filters and token lifetime policies.
6. Search the rest of your configuration for the old application ID `65a544d1-2825-432e-9083-d73f9c786d92`, and review each match.

Updating the devices

A device uses the new application after a firmware update to UniconOS 7.2611 or 7.2603.3000. Devices with an older UniconOS version keep using the old application until you update them.

Roll out the update in stages:

1. Update the devices of a pilot group.
2. Check that their users can sign in and start their resources, see Checking the change.
3. Update the other devices.

If you want to update devices before your tenant is ready, you can keep them on the old application for a while, see Setting the application ID on the device.

Keep the old application in your tenant, with its consent and its policies, until no device uses it any more. The sign-in logs of Entra ID show this: they no longer list sign-ins with the old application ID. Then remove the old application from your tenant.

What users notice

When a device moves to the new application, each user has to sign in once more with their Entra ID account. The sign-in that the device saved for the old application doesn't work with the new one.

If users sign in to UniconOS with Entra ID, the move also changes how UniconOS recognizes them, because Entra ID identifies a user differently for each application:

- A session that was locked before the move can't be unlocked after it. To get back in, the user logs off from the lock screen and signs in again.
- UniconOS treats the user as a new user and creates a new home directory.

Setting the application ID on the device

The application ID is a device setting, so you can change it without a new UniconOS version. The setting applies to the AVD client and to the Entra ID sign-in on the device. You don't need to set it: without the setting, both use the new application.

Set the application ID only to point devices to another application:

- To keep devices on the old application for now, for example when you update them before your tenant is ready. You can then move them to the new application group by group. This works only until the old application is retired.
- To use an application that you registered in your own tenant.

You set the application ID as an advanced file entry for an OU or a device. In UniconOS Management Board, you find advanced file entries in the **Advanced file entries** section of the advanced device configuration.

The advanced file entry has these values:

FILE	SECTION	KEY	VALUE
`/setup/terminal.ini`	`AVD`	`OauthClientId`	The application ID, for example `65a544d1-2825-432e-9083-d73f9c786d92` for the old application

The device uses the new value after it has received the entry and restarted.

Checking the change

On a device of your pilot group, check that the AVD client works with the new application:

1. Start the AVD client, and sign in if it asks you to.

The user's Azure Virtual Desktop resources are shown.

2. Start a resource.

   The session opens.

To see which application ID a device uses, run `ucsettingsctl -o AvdConfiguration` in a terminal on the device. In Entra ID, the sign-in logs list the sign-ins of updated devices under the new application ID.

If the consent or a sign-in fails, use this table to find the cause:

WHAT YOU SEE	CAUSE	SOLUTION
The sign-in fails. Users get no consent prompt, or a prompt that they can't complete.	Admin consent for the new application is missing.	Grant admin consent, see Granting admin consent.
The error `AADSTS650052` when you grant admin consent	The Azure Virtual Desktop service principal is missing in your tenant.	Register the resource provider `Microsoft.DesktopVirtualization` in your Azure subscription, and grant admin consent again.
The error `AADSTS50105` when a user signs in	The new application requires user assignment, and the user isn't assigned.	Assign the user, or a group of the user, to the new application.
The Microsoft sign-in page reports that the application with the given identifier wasn't found in the directory.	The device uses an application ID that doesn't exist, for example a mistyped `OauthClientId`.	Correct or delete the `OauthClientId` entry, and restart the device.

Security Considerations and Best Practices for running Citrix Unicon Scout

Security Considerations and Best Practices

Introduction

Threat Analysis

Modern IT systems face a range of security threats that often target communication paths between users and services. One major threat category includes man-in-the-middle attacks, in which an attacker secretly intercepts or manipulates communication between two parties. Such attacks can be carried out through techniques like network spoofing, ARP spoofing, the exploitation of unsecured networks, or by leveraging malware already present on a device.

Another significant threat involves backdoor and rootkit attacks, where attackers exploit hidden or poorly secured access points within a system. Once a backdoor is found or planted, a rootkit can be installed to gain persistent, unauthorized control over the system, making detection and removal difficult.

Closely related are viruses and Trojan horses, which typically rely on backdoors to infiltrate systems. Trojans disguise themselves as legitimate software, tricking users into installing them, while viruses spread across systems and networks. Social engineering, especially through deceptive emails, remains a common method for delivering these malicious programs to unsuspecting users.

Additionally, malware and ransomware pose a substantial risk. These programs are often installed intentionally by users or administrators who believe they are deploying legitimate tools. Hidden within them, however, is harmful code capable of encrypting data, disrupting operations, or opening additional attack pathways.

A further consequence of these attack types is data theft, a frequent outcome once a system is compromised. Stolen data may include sensitive corporate information, intellectual property, or personal data, and its exposure can significantly harm a company’s operations, reputation, and legal standing.

Security with eLux and Scout

Unicon's secure operating system eLux and the corresponding management software Scout are designed to minimize the potential threats and their impact. The security architecture includes the following features:

1 eLux does not store any personal information or sensitive information locally on a device. The system is kept read-only and only digitally signed software can be installed. This removes the risks for malware and ransomware, it also minimizes the possibility that a rootkit can be installed. 2 Enterprise communication offered by eLux uses state-of-the-art security measures to allow

certificate based IEEE 802.1X network authorization. Certificates are stored in TPM chips on the eLux device. All internal network communication between Scout and eLux is secured by TLS. Users are warned when they connect to insecure networks. This mitigates the risk for man-in-the-middle attacks accordingly. 3 Rights management allows the Scout administrator to granularly adjust individual rights for the users of eLux devices. This can also include USB device handling. This reduces the chances of data theft by insider threats. 4 Applications and the eLux system are sandboxed and shielded to minimize the risk to security breaches and backdoor attacks. Additional sandbox features are available for all user applications and browser content. 5 eLux is a secure Linux system that adheres to Linux security standards. CVEs are patched and packages are updated accordingly when necessary as well as in regular updates. 6 Further security measures are also supported by eLux like smartcard authentication, running browser applications in an encapsulated kiosk mode and more.

Layers of Protection

Security in eLux is based on several layers of protection. Even if one defense is breached, the multitude of layers allow to protect the system.

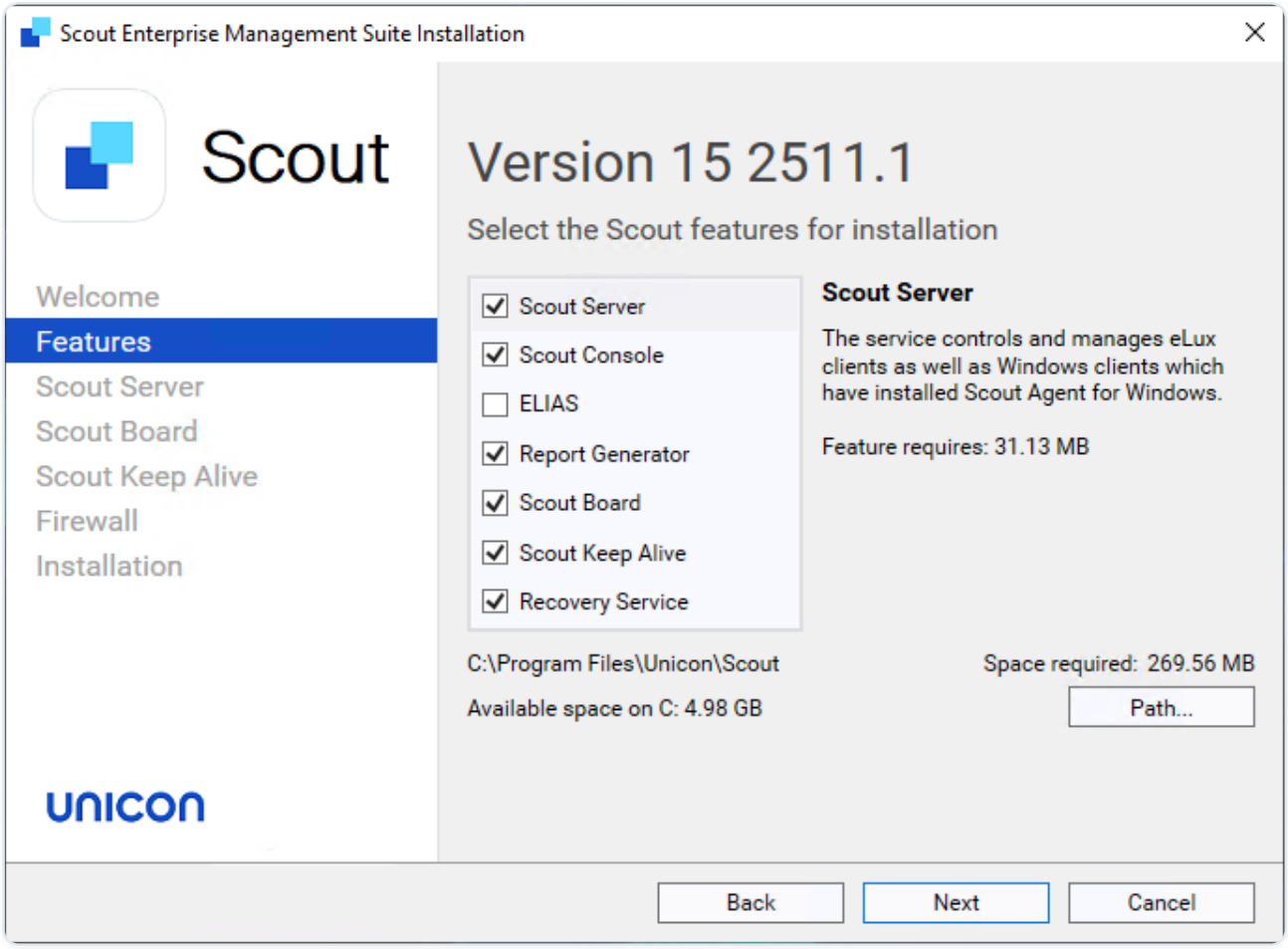
SECURITY LAYER	ADVANTAGES
Custom Linux kernel build with own kernel configuration	Only necessary drivers and components are enabled
Custom Linux kernel build with own kernel configuration	Unneeded code is not integrated, thus minimizing a possible attack vector
Secure network communication	Support of modern network security, IEEE 802.1X
Secure network communication	Encrypted communication with Scout to avoid man-in-the-middle attacks
Separation of privileged system access and unprivileged access	By default code will be executed unprivileged
Only a selected number of default components are installed	Components are hand selected from Ubuntu including Ubuntu backports of upstream security fixes
Only a selected number of default components are installed	Minimizing the number of packages, thus minimizing a possible attack vector

SECURITY LAYER	ADVANTAGES
Read-only filesystem	Programs and processes cannot change, write or delete files
Read-only filesystem	Avoids unauthorized and unwanted changes
Encrypted filesystem	Content of filesystem cannot be retrieved with external tools
Encrypted filesystem	Key is secured in TPM chip that recognizes tampering with the system
Individual customer software images	Allows administrators to fine tune installed packages
Individual customer software images	Minimizing possible attack vectors to only keep the essential files
Installation of only signed packages	Possibility to only allow software packages from trusted sources
Installation of only signed packages	Avoid man-in-the-middle attacks
Regular security updates for the current release and long term support	secure packages, where security fixes were published
Additional security packages and features available	For scalable security, additional security components are available in the form of firewall, network access control, sandboxes and more
Secure login procedures include Active Directory, smart card, identity providers	Protection against unauthorized access
Granular user rights granted by administrators	Administrators can control the rights of users very detailed to mitigate insider threats
No personal data is stored	Personal data and login data is never stored on the device
No personal data is stored	No data can be lost or compromised by the theft or loss of a device

Installation and Setup

Installation Scope

During the installation select *User defined* as installation scope. Only select the Scout features needed for your setup, unselect any features that are not necessary.



MS SQL Setup

Data at Rest

On the MS SQL setup ensure the encryption of the data at rest through Transparent Data Encryption (TDE). The performance impact through TDE should be neglectable for average and peak Scout database communication.

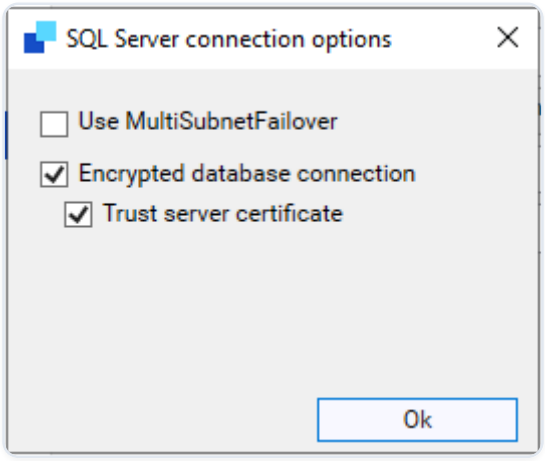
Transparent Data Encryption (TDE) encrypts the physical database files (MDF, NDF) and transaction logs (LDF) using AES or 3DES at the storage layer. This means the encryption happens on the server itself and is transparent to applications: data is encrypted before being written to disk and decrypted automatically when loaded into memory. Additionally, TDE protects backups, preventing offline attacks against copied database files or backup media. TDE uses a hierarchy of keys: a Database Encryption Key (DEK) stored in the database boot record, protected

by either a server certificate or an asymmetric key stored in an external key store such as an HSM. This prevents an attacker who obtains database files from reconstructing the plaintext content.

Data in Transit

MS SQL Server protects data in transit using industry standard TLS (Transport Layer Security). When TLS is enabled for a SQL Server instance, all communication between clients and the server —queries, results, login credentials—is encrypted before transmission across the network. This prevents network level attackers from capturing or modifying data exchanged between endpoints. TLS requires installing a trusted certificate on the SQL Server host and configuring SQL Server Network Configuration to enforce encryption.

In the Scout installation select *"Connection options..."* and select *"Encrypted database connection"*.



Scout and eLux

Data at Rest

The eLux system partitions are already mounted read-only. By default disk encryption should always be turned on. This is done by selecting the *"Encryption FPM"* in the BaseOS when configuring an eLux image. The key for the encryption will be stored safely in the TPM 2.0 chip. This ensures that all information stored on the device remains unreadable if the hardware is lost, stolen, or tampered with. TPM 2.0 provides a hardware-based root of trust, meaning the encryption keys are securely bound to the device and cannot be extracted or used elsewhere, which significantly raises the barrier for attackers.

Scout also encrypts the passwords that it stores in the Scout database, with a data key of the installation. Unless you keep this key in a key file, the key is part of the database and of every backup, see Data key for stored passwords.

Data in Transit

By default the data transfer between Scout and eLux is encrypted using at least TLS 1.2. To add additional trust the certificate should be verified when used. To ensure this, set the TLSVerifyOption according to the Scout Installation documentation, configure the correct trust level and certificates. TLS-encrypted communication is useful because it protects management traffic and device communication from interception, tampering, and impersonation. In the eLux and Scout environment, TLS ensures that configuration updates, commands, and status data exchanged between endpoints and the management server cannot be read or altered by unauthorized parties.

Connecting devices outside the corporate network should be done via a dedicated VPN setup to connect eLux with Scout or using the Scout Cloud Gateway (SCG). SCG establishes a built-in, certificate-based VPN tunnel for each endpoint, removing the need for traditional VPN solutions while still ensuring fully encrypted communication back to the Scout server. This allows remote, home-office, and mobile users to receive the same configurations, updates, and policies as on-premises devices with minimal user interaction. SCG also streamlines onboarding through automated authentication and certificate issuance.

Scout Server also checks the certificate of the Citrix® License Server. Keep the default: Scout Server then refuses a certificate that Windows doesn't trust and that isn't the certificate that Scout accepted when you configured the License Server, see Licensing.

Device Configuration

Network

Network settings rely on the given infrastructure. We recommend to set up eLux in an IEEE 802.1X network environment.

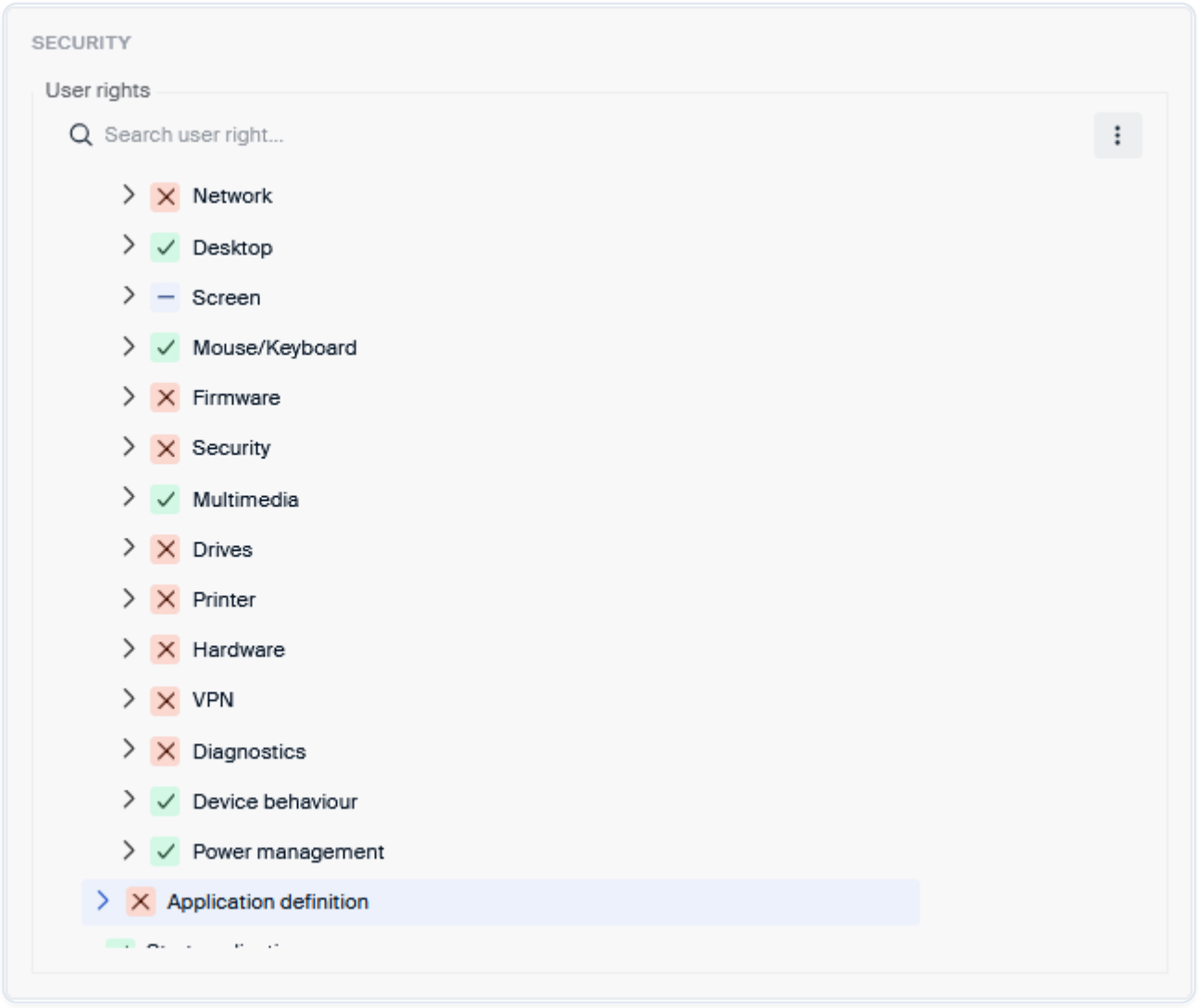
802.1X strengthens network security by ensuring that only authenticated and trusted devices can connect. Every connection attempt is validated before access is granted, which prevents unauthorized devices from joining the network and reduces the risk of internal threats. The protocol enables identity-based access control, enhances protection against malware and rogue devices, and supports a modern Zero Trust security approach. This results in a more secure, well-regulated, and resilient network environment across both wired and wireless infrastructures.

Restrict User Access

When configuring the device, in the *Security* tab the user rights need to be set. Restrict as much as possible to ensure that users do not tamper with any internal system settings. We recommend

restricting at least the following settings, these settings need to be set by the administrator on Scout:

- Network
- Screen Advanced
- Firmware
- Security
- Drives
- Printer
- Hardware
- VPN
- Diagnostics
- Application definition



User Authentication

Users should authenticate when using an eLux device. eLux supports a range of authentications, however, also a web-based or VPN-based authentication for a customer setup could be sufficient. This depends on the use-case and the customer environment.

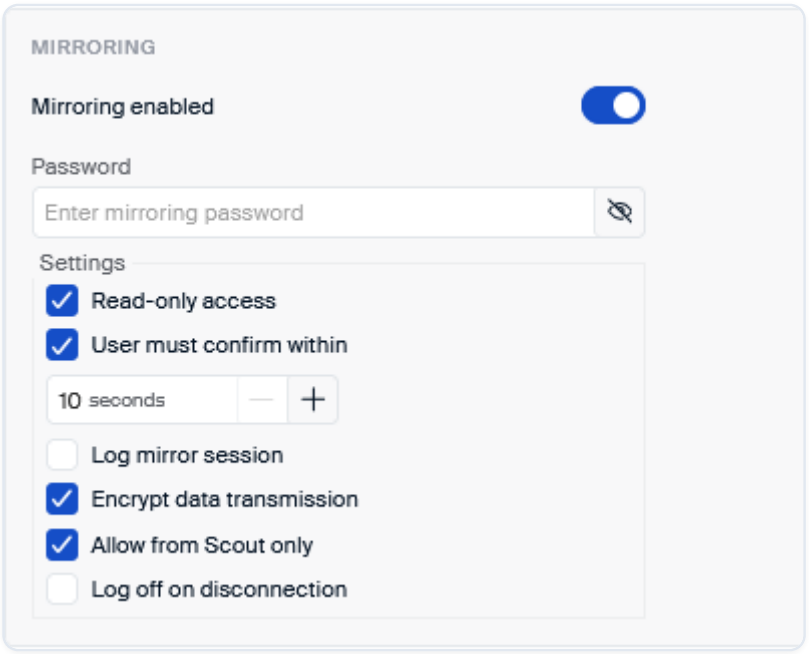
We recommend turning off *"Show last user"* when using one of the supported authentications on eLux. Showing the username of the last logged-in user is a security risk, especially on shared or publicly accessible systems. When eLux displays the last signed-in username, it gives an attacker half of the credentials needed to breach an account. This enables faster brute-force or targeted password-guessing attacks because the attacker no longer needs to guess the username.

Mirroring

Mirroring is a powerful tool in any helpdesk scenario, allowing the administrator to have direct access to the eLux device. To reduce any risks from man-in-the-middle attacks and inside threats we recommend to:

- Only allow the administrator read access to the target machine
- Ask the user to confirm with the mirroring of his device
- Allow access only initiated by this Scout server

UniconOS Management Board mirrors devices over a fully encrypted WebRTC connection.



Kiosk Mode

When setting up a kiosk mode for eLux, e.g. when used in a public setting, ensure to turn off *"Show last user"* in the settings. Use the automatic log-off feature to ensure no open session can be hijacked easily.

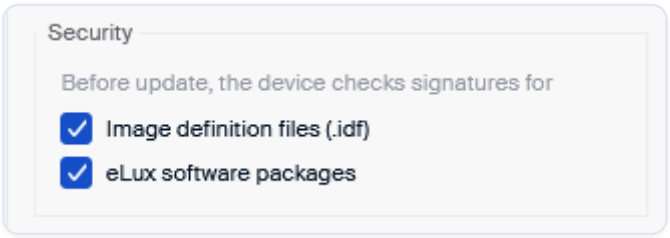
For a browser kiosk mode, customize the browser to remove the navigation bar and print menu if possible.

Firmware

For firmware updates, use secure web servers with HTTPS configured and username and password set for additional security.

Software in UniconOS Management Board allows the selection of features (FPM) on a very granular level. This should be used to minimize the image size to reduce the possible attack surface. For each package check the FPM selection and only select FPMs that are needed by the user.

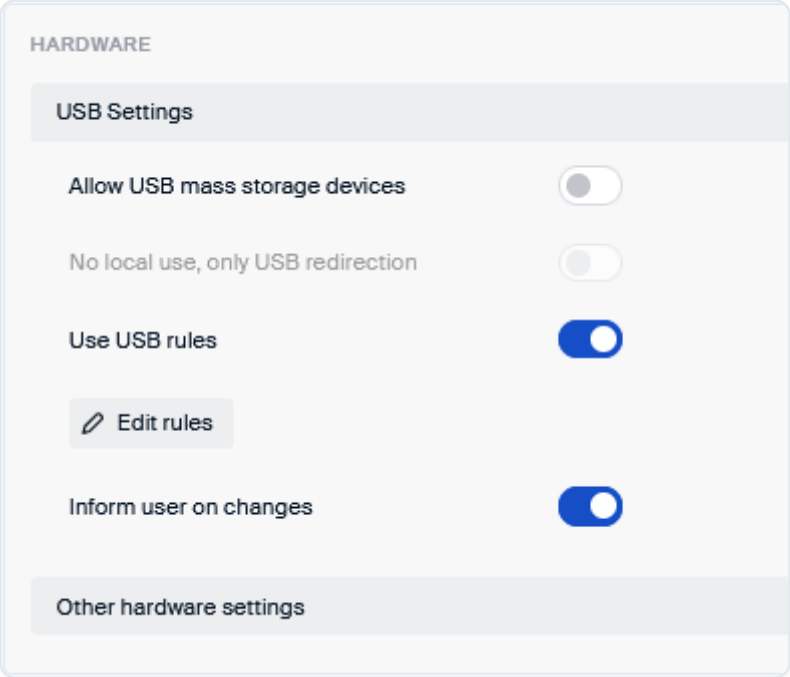
Do not install any packages from unknown origin. All packages by Unicon and Citrix are signed with the official private keys and can be checked with the default certificates of each container in **Software**, see Certificates. This should be enforced by checking both the image file (IDF) and the packages for valid signatures, see Checking signatures.



Hardware

Restricting or disabling USB device access on eLux is a strong security measure because it prevents untrusted removable media from introducing malware, leaking data, or bypassing the system’s controlled and centrally managed environment. This makes it possible to enforce USB blocking or read-only policies, which reduces exposure to malware, data exfiltration, and unauthorized device use.

We recommend disabling the use of USB mass storage devices. Instead, define USB rules to allow or deny access to specific USB device classes.



Diagnostics

Diagnostics should be used for trouble-shooting issues on eLux systems. On production systems it should be turned off to avoid accumulating log and system data in the temporary folders.

Sandboxing Applications

With the additional Firejail package local eLux applications can be run in an additional sandbox. This can be highly useful on eLux because it adds an additional, configurable isolation layer that strengthens the system's architecture. Firejail allows applications, especially browsers and other internet-facing tools, to run inside tightly restricted namespaces and profiles, limiting their access to the file system, network, and system resources. This containment helps prevent malicious web content or compromised applications from affecting the rest of the device, complements built-in browser sandboxing, and supports flexible security scenarios such as switching profiles when devices move between networks.

We recommend using Firejail for scenarios where the Built-in Browser is used in a mobile use-case often switching between safe and unsafe networks.

Administrator Permissions

Hardening administrative access in Scout is essential because the management server has complete control over all registered eLux endpoints, their firmware, configurations, certificates, policies, and operational behavior. A compromise of Scout administrator access would immediately translate into a compromise of the entire endpoint fleet. For that reason, administrator hardening

must be approached with strict least-privilege design, identity assurance, and configuration governance.

Scout supports multi-admin deployments and granular permissions, which must be used to separate operational responsibilities. Only a minimal number of individuals should receive super-admin privileges, as these roles can modify global configuration, certificate handling, onboarding flows, and security-critical settings. Delegated administrators should instead be restricted to narrow functional scopes such as device group configuration, OU-specific adjustments, or image deployment. It is recommended to give delegated administrators only access to predefined firmware configurations, predefined commands and command templates.

Scout supports authentication via identity providers like Entra ID and Okta using OIDC or SAML. These must be used in place of local accounts, enabling MFA enforcement, conditional access, centralized password policy, and identity lifecycle management. Relying on enterprise identity providers ensures that administrator access remains tied to corporate identity systems rather than isolated credentials that may not be governed by enterprise policies. From 15.2609, UniconOS Management Board also rejects a SAML response that was not issued for a sign-in it started itself, and accepts each response only once.

UniconOS Management Board ends a session after 30 minutes without activity, and 12 hours after the sign-in at the latest. Don't extend these limits more than you need, see Configuring session limits for UniconOS Management Board. After 10 failed sign-ins of an account from one address within 15 minutes, UniconOS Management Board refuses further sign-ins of this account from this address for up to 15 minutes.

UniconOS Management Board checks the rights of an administrator on every device and configuration that an action reaches, and a change of rights applies without a new sign-in, see Administrator rights in UniconOS Management Board.

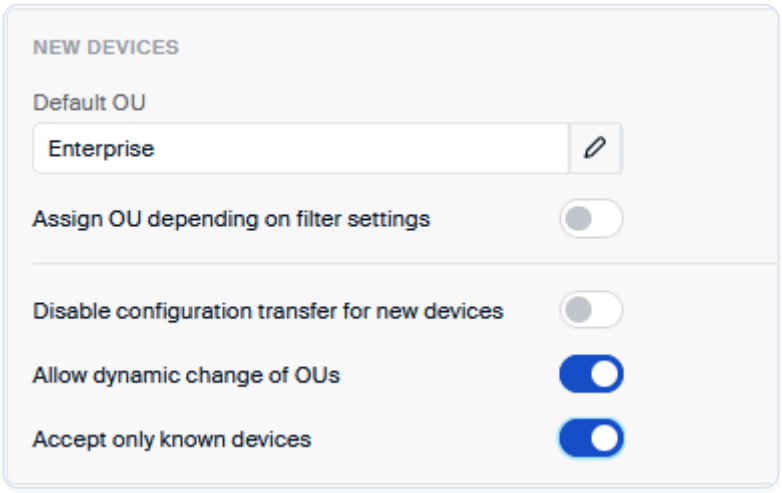
Scout Preferences

Devices

Setting a secure device password is important. The device password is used as a last line of defense to prevent man-in-the-middle attacks. It also is used to secure command execution with system rights on eLux. Therefore, only the master administrator should know the device password, other administrators should only use predefined commands if they need to run commands with system rights.

New devices should be sent to a quarantine OU with limited settings, e.g. no certificates, no applications. From there they can be moved to the right OU by the administrator. Only accepting only known devices further prevents malicious attacks where rogue eLux devices are connecting to Scout.

Protect the OUs whose settings not every device may get with an OU access password, and keep the setting **Devices that join a protected or hidden OU without sending its password to Scout Server** at **Reject**. Scout Server then places a new device that doesn't send the password in the default OU, and it keeps a device that moves itself without the password in its current OU. Place devices that can't send the password, for example devices that get their OU from DHCP options, with OU filter rules, see Devices.



Predefined firmware configuration

Firmware updates for eLux and UEFI BIOS should be prepared by the master administrator. All other administrators should only be able to select from the list of predefined firmware updates. This prevents administrators from installing untested or potentially dangerous images, even if they have no malicious intent. This prevents administrators from deviating from hardened configurations, which could otherwise introduce vulnerabilities or weaken compliance. Firmware-level control also ensures that only validated OS versions, UEFI update processes, and secure package sets are used, reducing risk from untested images or misconfigurations; this aligns with the roadmap focus on secure firmware handling, logging improvements, and controlled image/firmware update workflows in Scout. Together, these restrictions ensure device integrity, prevent unauthorized or accidental downgrades or insecure configurations, and maintain a robust, enforceable security posture.

Predefined commands

Predefined commands as discussed above allow the master administrator to define a set of commands that other administrators can send to the devices even with system access. This

prevents unauthorized, risky, or harmful actions from being executed on managed eLux endpoints. By limiting which remote commands an admin can send organizations minimize the risk of accidental misconfiguration, privilege misuse, or insider threats.

Command Templates

Command templates in Scout enhance security by tightly controlling how device-level actions can be executed, ensuring that administrators are limited to predefined, validated command structures rather than entering arbitrary or potentially harmful input. By providing centrally created command templates organizations ensure that only safe, vetted commands can be executed on devices, reducing the attack surface and ensuring consistent behavior across large deployments. This safeguards the environment from malformed parameters, enforces compliance with operational policies, and prevents delegated admins from performing unauthorized or risky low-level operations across the endpoint fleet.

Recovery Settings

We do not recommend to use PXE boot in general as it operates before security controls are active, uses historically insecure protocols (DHCP/TFTP), and is susceptible to rogue DHCP/TFTP servers, man-in-the-middle image manipulation, remote code execution vulnerabilities in UEFI PXE stacks and unauthorized image capture or injection. Use PXE boot only on trusted networks with strong security controls on the infrastructure level. However, when needed, PXE must only be used on trusted, isolated networks with strong security controls at the infrastructure level.

Short guides

A short guide provides information on specific procedures for our products that goes beyond the administrator's guide. Usually, short guides can be followed without having to read the full administrator's guide.

Short guide example topics are the migration between two versions of our operating system eLux® and recovery procedures for eLux devices.

All contents from the uDocs online documentation are also available for download in the PDF downloads section.

Introduction to Installation

System requirements

> **Note:**

> The following information refers to Scout Enterprise Management Suite® 15.x.

Requirements for Scout Server

- Hard disk space 600 MB (only Scout Enterprise Management Suite, the software container requires additional space)
- Microsoft Windows Server 2019, 2022 or 2025

each including the relevant software updates provided by Microsoft at the time of installation

- Citrix® License Server 2607 (build 56200) or later
- Microsoft .NET Framework version 3.5 and Microsoft .NET Framework version 4.5.1 or later
- Suitable ODBC driver

> **Note**

Please use the latest GA version of the Microsoft ODBC driver, as more recent versions might need the latest drivers. Scout looks for the ODBC driver in this sequence:

- ODBC Driver 18 for SQL Server (since 15.2309, 2023)
- ODBC Driver 17 for SQL Server (since 15.9.1000, 2020)
- ODBC Driver 13 for SQL Server (since 15.9.1000, 2020)
- ODBC Driver 11 for SQL Server (since 2015)
- SQL Server Native

One of the drivers must be installed. Setup itself needs ODBC Driver 17 or 18 for SQL Server, and it stops if neither is installed.

Requirements for the database system

- Microsoft SQL Server 2019, 2022
- or for small installations: MS SQL Server Express

Requirements for Software Management

- Microsoft Windows Server 2019, 2022 or 2025 for the Software Management backend, on the UniconOS Management Board computer or on a computer of its own
- If the backend runs on a computer of its own, the clocks of this computer and of the UniconOS Management Board computer must not differ by more than one minute, see Software Management backend on another computer
- At least 20 GB free disk space for the data folder of the backend, ideally on a disk or partition of its own. The data grows with every UniconOS version that you import, by about 8 GB for an AllPackages archive with its recovery image. While an archive is imported, the drive needs free space of about three times its size.
- A data folder that setup accepts: not the root of a drive, belonging to Administrators, without other data and without a link or junction on its path, see Installing Software Management
- A web server or an FTP server that publishes the container folders to the devices and reads them on every request, see Publishing a container to devices

Requirements for the web applications UniconOS Management Board and Scout Cloud Gateway

See also system requirements for Scout Cloud Gateway in the SCG guide

- Web browser / minimum version
  - Mozilla Firefox ⇒ 136
  - Google Chrome ⇒ 137
  - Microsoft Edge ⇒ 127

> **Note:**
>
> Some functions use pop-up windows. Make sure that pop-up windows are not blocked by your browser. This function can usually be found in the browser settings under **Privacy** or **Security**.

- Screen resolution Full HD

For support periods and the compatibility matrix, see the whitepaper **Releases, Lifecycles and Compatibility**.

System limitations

There are no system restrictions known for any component of the Scout Enterprise Management Suite.

You can run additional services such as Citrix XenApp on the same system.

Database support

Scout requires database software such as Microsoft SQL Server.

Overview

Authentication to SQL Server

SQL Server users and application roles

Data key for stored passwords

Scout Server cluster

Number of ODBC connections

Overview

Microsoft SQL Server

You can use any Microsoft SQL Server version with available product support as an SQL database. We recommend that you create the required databases before installing Scout Enterprise Management Suite®.

A minimum of one **Scout database** is required to manage the following data:

- Device configuration
- Application definitions
- Asset data (static)
- Server settings
- Administrator management
- License information
- Transaction logging

A Scout database requires about 50 MB free disk space per 1.000 devices.

For Scout versions up to Scout 15 2204, the Scout Statistics Service was included as an optional component of the Scout Enterprise Management Suite which required a dedicated statistics database. From Scout 15 2209, the Scout Statistics Service has been replaced by the Scout Keep Alive service. The keep alive data are now stored in the Scout database.

New databases can alternatively be created in Microsoft SQL Server during the installation process of the Scout Enterprise Management Suite, provided the required permissions are available.

Changing the database connection

To change the database connection of Scout Server, use the Scout Database Connection Editor. For further information, see Database connections.

Stored passwords

The Scout database also holds the data key that Scout uses to encrypt stored passwords, unless you keep the key in a key file. A database backup therefore contains this key. For further information, see Data key for stored passwords.

Database cleanup

To delete outdated data from the Scout database, use **Database cleanup** in UniconOS Management Board: in the **Navigation** menu, click **Preferences**, and then click **Database cleanup**.

Authentication to SQL Server

When you select the **Microsoft SQL Server** database type during installation, you can choose between two authentication methods for the database engine, **Windows authentication** and **SQL Server authentication**.

The authentication requires either an SQL user or a Windows user. Each of them must be a member of certain database roles in SQL Server. For further information, see SQL Server users and application roles.

METHOD	DESCRIPTION
Windows authentication	'Trusted connection': The user identity is confirmed by Windows.
	The Scout service must be run with a user account equipped with the relevant permissions in SQL Server (For Scout Enterprise Management Suite 15.5 and later versions, gMSA (Group Managed Service Accounts) are also supported).
	In the Scout installation dialog, specify the account name in the form `DOMAIN\username` (no case-sensitivity) Example: `INT\mmi`
	Note: For the Scout installation, either a user with the relevant permissions in SQL Server or the technical user account (Log on as a service) to be used must be logged on to the Windows server system.
SQL Server authentication	The username and password must refer to an SQL Server user account with the relevant user rights in SQL Server.

METHOD	DESCRIPTION
	In the Scout installation dialog, specify the SQL username and password.

If you choose to install the Scout Enterprise Management Suite® completely, three databases are required. You can configure access to the three databases in the following ways:

- If the same authentication method is used for all databases, you can configure different users for each database.
- If the same user is used for all databases, you can configure different authentication methods for accessing each database.

SQL Server users and application roles

Below description gives a rough overview of the permissions required in SQL Server, as well as how to use an application role. An application role increases security by assigning operational administrators only the necessary permissions for the duration of a session.

Create databases

We recommend that you create the required databases in SQL Server before installing the Scout Enterprise Management Suite®.

Please note the following:

- To create databases in SQL Server, you need the SQL Server role **dbcreator** as a minimum.
- The database names may be freely chosen.
- The tables within the databases are created by the installation routine of the Scout Enterprise Management Suite.

> **Note:**

> You can restore a backup of the Scout database on another SQL Server or under a new name. Scout Server binds the license data to the restored database by itself when it starts, so you don't need to unlock the database. For further information, see Restoring or copying the Scout database.

> If you keep the data key for stored passwords in a key file, copy the key file to the new server as well, see Data key for stored passwords.

Permissions for users of the Scout Enterprise Management Suite

SQL SERVER ROLE

For the use of the Scout Enterprise Management Suite, the server role **public** in general is sufficient. Only if users are to perform additional tasks in SQL Server, extended permissions are necessary. For example, restoring databases requires the **dbcreator** server role.

SQL DATABASE ROLE

On database level, for console-only users in standard environments, the database roles **db_datareader** and **db_datawriter** are sufficient. **db_owner** is needed, for example, for database activities related to updating to a new Scout version. The same applies for performing any configuration and maintenance activities on the database.

The users must be mapped to the default schema dbo.

> **Note:**

> For environments with SQL Server clusters, additional permissions such as VIEW SERVER STATE and VIEW ANY DEFINITION are required.

SQL application role

In the case of Windows authentication for SQL Server, users are authorized to also log on to the SQL Server Management Studio. For scenarios like this we recommend using a dedicated user group with limited database rights for console-only users.

Permissions to access SQL Server tables can additionally be controlled and restricted for all databases via a system-wide SQL application role. The name and password of the application role must be defined in the relevant database.

A console user for SQL Server then only needs the database role **public** in order to execute the stored procedure for activating the application role. Once the application role is active, the connection to SQL Server loses the user permissions and assumes the permissions of the application role. Other databases in which **guest** has been disabled will be inaccessible to the application role. The application role permissions remain active for the duration of a session.

Configuring console users and an application role in SQL Server Management Studio

The following instructions refer to the Scout database.

1. In SQL Server Management Studio, under **Security > Logins**, add a new user group (Example: **Console users**) with the following options:

Server role	Public
User mapping > Database role membership	db_datareader
Securables	Set non-required permissions to Deny.Example on server level: **Alter any database** > Deny

2. Below the Scout database, under **Security > Schemas**, add a new schema (Example: **Console users schema**). Under **Schema owner**, select the **Console users**.
3. Below the Scout database, under **Security > Roles > Application roles**, add an application role (Example: **Console users role**) with the following options:

Default schema	Console users schema
Password	Choose freely
Securables	Select your Scout database and set all permissions except **Connect** to Grant

Defining the application role in the Scout database

1. To specify the application role data in encrypted mode, in a first step, encrypt the name and password of the role. For further information, see Encrypting values.
2. Back in SQL Server Management Studio, for the Scout database, edit the **System** table.
3. Add one row for the name and one row for the password of the application role. Paste the encrypted values into the table:

SYSTEMID	PARAMNAME	PARAMVAL
…		

SYSTEMID	PARAMNAME	PARAMVAL
< n>	`RNameEx`	*< encrypted name of application role>*
< n>	`RPassEx`	*< encrypted password of application role>*

When the Scout Console is started, the fields are read and the access rights of the application role are set.

Data key for stored passwords

Scout stores some passwords in the Scout database, for example the passwords in configuration labels, the kiosk lock password and the password for automated logon to Active Directory. From UniconOS Management 2611, Scout encrypts these passwords with a key that belongs to your installation alone: the data key. This page explains where Scout keeps the data key, what this means for database backups, and how you can keep the key apart from the database.

Unless you use a key file, Scout creates the data key when Scout Server starts for the first time with version 2611, and stores it in the Scout database. Every time Scout encrypts a password, the result is different, even for the same password.

Database backups contain the data key

A backup of the Scout database contains the data key together with the encrypted passwords. Anyone who has the backup can therefore decrypt the passwords. Protect your database backups as carefully as the database itself.

If a backup alone must not be enough to decrypt the passwords, keep the data key in a key file, see Keeping the data key in a key file.

Keeping the data key in a key file

Instead of keeping the data key in the database, Scout can read it from a key file on each computer that runs Scout Server. The key file is optional. Scout Server and the Scout Board Database Layer service read the file `%ProgramData%\Unicon\Scout\keys\data.key`. To use another path, set the environment variable `SCOUT_DATA_KEY_FILE` to the full path of the file. A computer that runs only UniconOS Management Board doesn't need the key file.

Create the key file on every computer that runs Scout Server with this database before Scout Server or the UniconOS Management Board Database Layer starts there for the first time with version 2611, that is, before you install or upgrade. Then Scout never stores a data key in the

database. If you add the key file later, Scout encrypts new passwords with the key from the file. The data key in the database stays there, because Scout needs it to read the passwords stored until then. These passwords stay readable with a database backup until you enter them again.

FORMAT OF THE KEY FILE

The key file is a text file with one key per line. Each line has the form `<key ID>:<key>`:

- The key ID has 1 to 16 characters: lowercase letters, digits, hyphens and underscores.
- The key consists of 32 random bytes in Base64 encoding. It has 44 characters and ends with `=`.

The first line holds the active key, which Scout uses to encrypt every new password. The lines below it hold earlier keys. Scout uses them only to read the passwords that were encrypted with them. Save the file as UTF-8, with or without byte order mark, or as UTF-16.

CREATING THE KEY FILE

1. On a computer that runs Scout Server, start Windows PowerShell as administrator.
2. Create the folder and a key file with a new random key. In this example, the key ID is `key1`:

```
$path = "$env:ProgramData\Unicon\Scout\keys\data.key"

New-Item -ItemType Directory -Force -Path (Split-Path $path)

$bytes = New-Object byte[] 32

[System.Security.Cryptography.RandomNumberGenerator]::Create().GetBytes
($bytes)

Set-Content -Path $path -Encoding ASCII -Value ("key1:" +
[Convert]::ToBase64String($bytes))
```

3. Set the permissions of the file so that only administrators and the account that runs Scout Server and the Scout Board Database Layer service can read it.
4. Copy the file to the same location on every other computer that runs Scout Server with this database.
5. Keep a copy of the file in a safe place, apart from your database backups.
6. If Scout Server already runs, restart Scout Server and the Scout Board Database Layer service on every computer, so that they use the key file.

WHEN THE KEY FILE CAN'T BE USED

Scout Server and the Scout Board Database Layer service stop right after they start if they can't use the key file:

- The key file is missing, although Scout has already used a key file with this database. So deleting the file doesn't switch Scout back to a data key in the database.
- The key file can't be read, or a line in it doesn't have the right format.
- The key file holds none of the keys that the database uses.
- A key in the file has a key ID that the database knows, but a different value.

To find the cause, check these places:

- `eluxd.log`, the log file of Scout Server, names the cause in a line that starts with `Scout Server cannot start: the key for stored secrets could not be loaded`.
- In the Windows Application log, an event from the source Scout Enterprise reports that Scout Server stopped with exit code 1018.
- The Scout Board Database Layer service writes an error to the Windows Event Log: `Scout Board DBLayer cannot start: the key for stored secrets could not be loaded.`

In the System log, Windows shows the exit code 1018 together with the text of an unrelated Windows error. Ignore that text. To solve the problem, put the right key file in place, and then start both services again.

Without the key file, Scout Server and the Scout Board Database Layer service don't start, and nobody can decrypt the passwords that Scout encrypted with its keys. So keep a copy of the file in a safe place. Once Scout has used a key file with a database, you can't switch this database back to a data key in the database.

Restoring the database on another server

You can restore a backup of the Scout database on another server, or copy the database there. What you need to do for the stored passwords depends on where the data key is:

- Without a key file, you don't need to do anything. The data key is part of the database.
- With a key file, also copy the key file to the new server, before Scout Server starts there. If you use `SCOUT_DATA_KEY_FILE`, set this variable on the new server too.

For the license data of a restored database, see Restoring or copying the Scout database.

Several Scout Servers on one database

If several computers run Scout Server with the same Scout database, they all share the data key of this database. Without a key file, there is nothing to copy.

With a key file, every computer that runs Scout Server needs the same file. Setup doesn't copy it. Put the file on a computer before you install or upgrade Scout Server there. A computer without the file can start neither Scout Server nor the Scout Board Database Layer service.

To upgrade such an installation to version 2611, stop Scout Server on every computer before you upgrade the first one. For all steps, see Several Scout Servers on one database. At its first start, Scout Server 2611 converts the stored passwords to a format that version 2609 and earlier can't read. For more information about this setup, see Scout Server cluster.

Changing the data key

If you use a key file, you can replace the active key with a new one. Change the key in two rounds, so that every computer can read passwords with the new key before any computer encrypts with it:

1. Create a new key with a new key ID, for example `key2`. These Windows PowerShell commands show a new key line that you can copy:

   ```
   $bytes = New-Object byte[] 32

   [System.Security.Cryptography.RandomNumberGenerator]::Create().GetBytes
   ($bytes)

   "key2:" + [Convert]::ToBase64String($bytes)
   ```

2. On every computer that runs Scout Server, open the key file in an editor that runs as administrator. Add the new key line below the active key, and save the file. Then restart Scout Server and the Scout Board Database Layer service on this computer.
3. On every computer that runs Scout Server, move the new key to the first line, and keep the earlier key below it. Then restart Scout Server and the Scout Board Database Layer service on this computer.

Afterwards, the key file looks like this:

```
key2:<new key>

key1:<earlier key>
```

Keep the earlier key in the file. Scout needs it to read the passwords that were encrypted with it.

Passwords from earlier versions

At every start, Scout Server converts the passwords that earlier versions stored:

- It encrypts label passwords and kiosk lock passwords that are still in the format of earlier versions with the data key. The passwords themselves don't change.
- It repairs label passwords that earlier versions saved with an older key or encrypted twice. Devices that use such a label get the repaired password with their next configuration update.
- It protects the passwords for automated logon to Active Directory with the data key. Devices get the same password as before. If Scout Server can't read the stored password for automated logon to Active Directory, it sends automated logon turned off and without a password, and `eluxd.log` names the device.

For each kind of password, Scout Server writes a summary line to `eluxd.log`. The lines start with `Label secrets:`, `Kiosk lock passwords:` and `AD auto-logon passwords:`. If Scout Server can't read a label password, it logs the label and the field, but not the password, and leaves the password as it is. Scout Server doesn't send such a password to the devices: it leaves out the label setting that holds the password, for example the application, the network profile, the network drive or the automatic logon of the label, and logs it. The rest of the label still reaches the devices, and for the setting that is left out, the device configuration applies. Enter the password again in the label.

From version 2611, a kiosk lock password can have at most 217 bytes in UTF-8 encoding, and UniconOS Management Board refuses a longer one. Scout Server can't convert a longer kiosk lock password from an earlier version, and the `Kiosk lock passwords:` line counts it as too long. Set a shorter kiosk lock password in the device configuration.

At its first start, Scout Server 2611 deletes the pass-through passwords that the **Kiosk Mode** section of the device configuration stored in earlier versions. Devices never used the settings of this section, and UniconOS Management Board no longer offers it. The deletion can't be undone.

Version 2609 and earlier versions can't read the new format. Don't start a Scout Server of an earlier version with the database again. Don't use the legacy Scout Console of an earlier version with the database either: if you save the settings for automated logon to Active Directory there, devices get a password that they can't use. To go back to an earlier version, restore the database backup that you made before the upgrade.

A later version will no longer read passwords in the format of earlier versions. Before you upgrade to a later release, let Scout Server 2611 start at least once. Then set again each password that the summary lines report as unreadable or too long.

Scout Server cluster

If you use an SQL database, several Scout Servers can connect to the same Scout database concurrently. Concurrent Scout Servers enable failure load balancing as well as the option to configure load balancing (ManagerLoadBalancing).

Each server of the cluster runs a complete installation: Scout Server, the Scout Board Database Layer and UniconOS Management Board. The servers share only the Scout database. There's no main server: every server is active, so administrators can work in the UniconOS Management Board of any server. If a server fails, its UniconOS Management Board isn't available either, so use the UniconOS Management Board of another server. From UniconOS Management 2611, the servers set up their connections by themselves, see What Scout sets up by itself.

All Scout Servers that use the same Scout database share its data key for stored passwords. If you keep the data key in a key file, every Scout Server needs the same file, see Data key for stored passwords.

Client devices that connect to a Scout Server receive a list of all currently running servers that access the shared Scout database.

FailureLoadBalancing

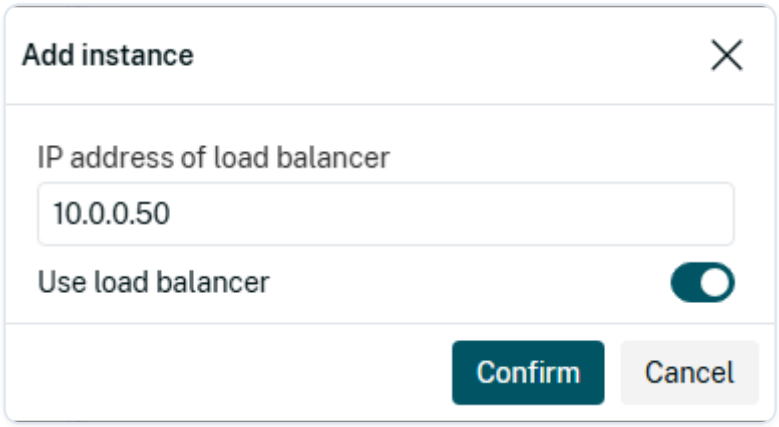
On start-up, a device tries to connect to the Scout Server it was connected to last time. If, however, that server is not available, it connects to the next server from the servers list which, subsequently, becomes the server the client tries to access by default from then on.

The FailureLoadBalancing mechanism restarts as soon as the client fails to connect to the same Scout Server.

ManagerLoadBalancing by dedicated load balancer

To define a dedicated load balancer, predefine the preferred manager address (IP address or name) for load balancing you want the devices to connect to:

- In the **Navigation** menu of UniconOS Management Board, click **Preferences**, and then click **UniconOS Management instances**. Click **Add instance**, enter the address in **IP address of load balancer**, turn on **Use load balancer**, and click **Confirm**.



The load balancer entry refers to an existing load balancer pointing to the relevant Scout Server. The load balancer entry allows you to assign devices to a particular Scout Server without changing the Scout configuration.

The load balancer name is evaluated by the devices on each client restart.

Procedure:

- The device restarts
- The device connects to the load balancer and then is forwarded to the appropriate Scout Server

If, however, the Scout Server identified by the DNS entry `ManagerLoadBalancer` is not available, the FailureLoadBalancing mechanism described above is used and the client accesses the next server on the list.

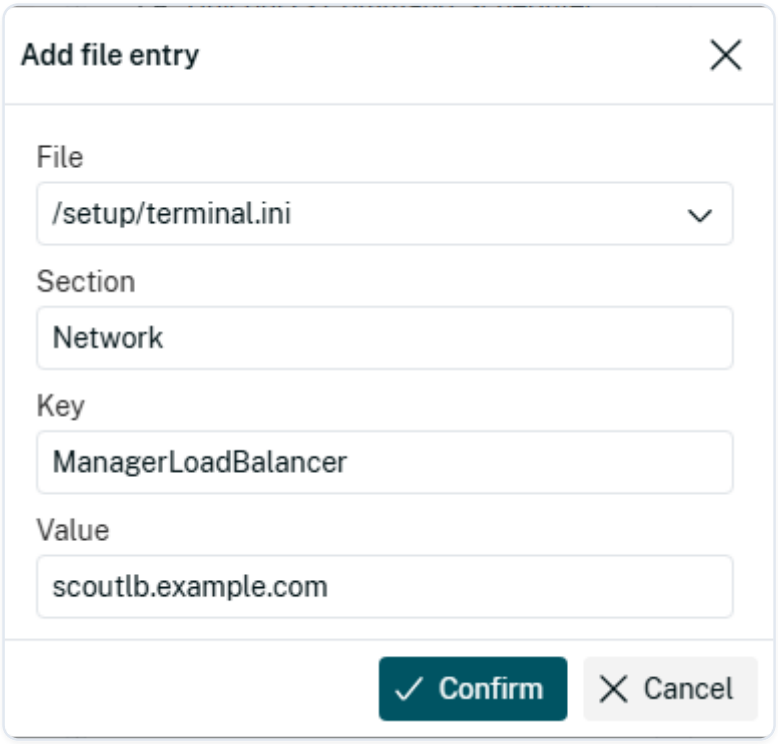
ManagerLoadBalancing by DNS entry

Alternatively, to predefine a preferred Scout Server you want the devices to connect to, you can use a DNS entry that you refer to in the advanced file entries.

- In UniconOS Management Board, add an advanced file entry for all devices, for an OU or for a single device, under **Advanced device configuration** > **Advanced file entries**:

FILE	`/SETUP/TERMINAL.INI`
Section	`Network`
Key	`ManagerLoadBalancer`
Value	*< FQDN DNS entry>*

For further information, see Advanced file entries.



`ManagerLoadBalancer` refers to a DNS entry pointing to the relevant Scout Server. In a separate step, the DNS entry must be defined on the DNS server. The DNS entry allows you to assign devices to a particular Scout Server without changing the Scout configuration.

The `ManagerLoadBalancer` parameter is evaluated by the devices on each client restart.

Procedure:

- The device restarts
- DNS entry `ManagerLoadBalancer` is resolved
- The device connects to the appropriate Scout Server

If, however, the Scout Server identified by the DNS entry `ManagerLoadBalancer` is not available, the FailureLoadBalancing mechanism described above is used and the client accesses the next server on the list.

What Scout sets up by itself

Installing, upgrading, repairing or modifying a server, or adding a new one, no longer breaks the connections between the servers. From UniconOS Management 2611, Scout takes care of these points:

- **One Scout Server key for all servers.** Scout Server uses a key to secure the events that it sends to the Database Layer. The Scout database holds this key, and every Scout Server and every Database Layer uses the copy in the database. Setup no longer replaces the key: an upgrade, a repair or a modification keeps it, and a new server takes it from the database.

- **Every UniconOS Management Board finds every server.** Every 60 seconds, UniconOS Management Board reads from the Scout database which Database Layers are running, and it receives the events of each of them. When the Database Layer of a server stops, UniconOS Management Board drops its connection to this server after about 2.5 to 4 minutes. If Scout Server still runs there, **Node health** then shows the server with a warning. You don't need to list the servers in the settings of UniconOS Management Board.
- **The settings of UniconOS Management Board stay.** An upgrade and, from 2611, also a repair keep your entries in the `.env` file of UniconOS Management Board.
- **UniconOS Management Board shows the health of every server. Node health** shows for each server whether this UniconOS Management Board receives its events, see Checking the health of the servers.

If the servers of your cluster stopped receiving each other's events because an earlier setup run replaced the key, the upgrade to UniconOS Management 2611 repairs the keys without further steps. Only the list of hosts that each Database Layer accepts still needs your attention, see Connections between the servers.

Connections between the servers

A job that an administrator starts in UniconOS Management Board, for example a command or a mirroring session, isn't tied to a server. The Scout Server that picks up the job first runs it, and the result comes back through the Database Layer of this server. Therefore, every UniconOS Management Board needs the events of every server.

Each UniconOS Management Board sends its requests to the Database Layer on its own server, on port 22150. It receives the events of all servers from their Database Layers, on port 22151. On every server, port 22151 must be open for all other servers.

From UniconOS Management 2611, each Database Layer also checks the IPv4 address of every computer that connects to it, see Configuring the Database Layer client check. A new installation allows only the server itself, and an upgrade adds only the servers that it finds connected. On every server, the list must contain `self` and the IPv4 addresses of all other servers. For example, if the servers have the addresses `10.0.0.11`, `10.0.0.12` and `10.0.0.13`, the `[Network]` section of `dblayer.ini` on `10.0.0.11` needs these settings:

```
[Network]

DBLayerClientCheck=Enforce
```

```
DBLayerAllowedClients=self,10.0.0.12,10.0.0.13
```

Without these entries, a server refuses the UniconOS Management Boards of the other servers. These UniconOS Management Boards then miss the events of this server: they show status changes of devices only after a reload, and mirroring sessions can fail.

Keep these points in mind:

- Give the servers static IPv4 addresses. The list can't hold host names.
- If a server has several network adapters, list the address that this server uses to connect to the other servers.
- If you change the address of a server or remove a server, update the list on all other servers.
- If setup created the firewall rules with the client check set to `Enforce`, the rules **Scout DBLayer** and **Scout DBLayer Publish** admit only the hosts in the list. With only `self` in the list, the firewall drops the connections of the other servers, and the Database Layer doesn't log them.
- If you can't keep the list up to date, set `DBLayerClientCheck=Audit`. The Database Layer then accepts every computer and only logs the computers that aren't in the list. In this mode, the client check no longer protects ports 22150 and 22151.

SETTINGS OF UNICONOS MANAGEMENT BOARD FOR THE OTHER SERVERS

UniconOS Management Board finds the other servers by itself. To edit the `.env` file of UniconOS Management Board, start the editor as administrator, and restart the **Scout Board** service after a change. These two settings in the file control which Database Layers UniconOS Management Board receives events from:

SETTING	WHAT IT DOES
`ZERO_MQ_ADDRESS_PUBLISH`	Lists Database Layers that UniconOS Management Board receives events from in addition to the servers that it finds, as a JSON list, for example `["tcp://localhost:22151"]`. Keep the value that setup wrote, and don't delete the line. Remove the entries of servers that no longer exist: while UniconOS Management Board can't resolve a name in this list, it adds no new servers.

SETTING	WHAT IT DOES
`ZERO_MQ_PUBLISH_DISCOVERY`	`false` stops UniconOS Management Board from finding the servers. `ZERO_MQ_ADDRESS_PUBLISH` must then list the Database Layers of all servers, including its own, for example `["tcp://localhost:22151","tcp://10.0.0.12:22151"]`. Without this setting, UniconOS Management Board finds the servers.

Adding a Scout Server

To set up a cluster, install the first server as usual. Then add each further server:

1. On the new server, install Scout with the Scout database of the cluster, see Installing Scout Enterprise Management Suite. Select the existing Scout database, and don't create a new one. Use the same database account, service account, License Server and UniconOS Management Board sign-in settings as on the other servers.

   In a silent installation, pass `/v"UCPROP_DBCREATE=0"` together with these settings, see Unattended installation.

   If another server of the cluster runs Software Management already, select the `User-defined` installation scope and clear **Software Management** and **Software Management backend**. Software Management keeps its containers in a data folder on its own server. If several servers run it, each of them lists the containers of the others but can't use them. Manage the software in the UniconOS Management Board of the server that runs Software Management.

2. On the **Firewall** page, under **Allow only these hosts to connect to DBLayer (ports 22150 and 22151)**, add the IPv4 addresses of all other servers after `self`, for example `self,10.0.0.11,10.0.0.12`. To let setup create the firewall rules, select **Automatically**.

   In a silent installation, pass the list in `DBLAYERALLOWEDCLIENTS`. To let a silent installation create the firewall rules, pass the switches and all port numbers, because a silent installation has no default ports. For example:

   ```
   /v"DBLAYERALLOWEDCLIENTS=self,10.0.0.11,10.0.0.12"

   /v"UCPROP_FIREWALL_ADD_RULES_SCOUTSERVER=1" /
   v"UCPROP_FIREWALL_SCOUTSERVER_PORT=22125"
   ```

```
/v"UCPROP_FIREWALL_DBLAYER_PORT=22150" /
v"UCPROP_FIREWALL_DBLAYER_PUBLISH_PORT=22151"

/v"UCPROP_FIREWALL_ADD_RULES_SCOUTBOARD=1" /
v"UCPROP_FIREWALL_SCOUTBOARD_PORT=22160"
```

3. On every other server, add the IPv4 address of the new server to the list of allowed hosts in one of these ways:

   - Run setup again and choose **Modify**, see Changing the hosts that may connect to DBLayer.
   - Edit `DBLayerAllowedClients` in `dblayer.ini`, and restart the **Scout Board Database Layer** service, see Client check settings. If the firewall rule **Scout DBLayer Publish** of this server admits only certain addresses, add the new server to the rule as well.

4. Open **Node health** in the UniconOS Management Board of every server, see Checking the health of the servers.

   Within about two minutes, every UniconOS Management Board lists the new server with the status **OK**. During the first hour, **Last event received** can show **Unknown**.

You don't need to copy a key or to change the settings of UniconOS Management Board on any server.

Upgrading the servers

Upgrade the servers one after the other. An upgrade keeps the Scout Server key in the Scout database and the settings of UniconOS Management Board, so the servers keep working together. For the upgrade to UniconOS Management 2611, keep to the order in Several Scout Servers on one database, because this upgrade converts the stored passwords and sets up the list of allowed hosts.

Checking the health of the servers

In UniconOS Management Board, **Node health** shows every server of the cluster and checks whether this UniconOS Management Board receives the events of each server. In the **Navigation** menu, click **Preferences**, and then click **UniconOS Management instances**. Each UniconOS Management Board shows only its own connections, so check **Node health** in the UniconOS Management Board of every server. For the columns and their results, see UniconOS Management instances.

For monitoring tools, the health check of the REST API returns the same information:

```
GET https://<scout-board-host>:<port>/rest/api/v1/healthcheck?scoutNodes
```

The request needs a signed-in administrator, see Using the REST API. The answer contains the section `response.scoutNodes`, with one entry per server in `nodes`. The overall result is in `response.scoutNodes.status.result`: `0` means OK, `1` failing, `2` warning, and `3` that not all information is available. The `message` at the top of the answer is `OK` whenever the health check itself worked, also if a server fails.

The log files show more details. By default, both are in `%PUBLIC%\Documents\Unicon\Scout\ScoutBoard\`:

- `backend.log`, the log file of UniconOS Management Board, shows `ZeroMQ discovery: subscribed to tcp://<address>:22151 (<name>)` when UniconOS Management Board adds a server, and `ZeroMQ discovery: dropped` when it drops one.
- `dblayer.log`, the log file of the Database Layer, shows a warning that starts with `Warning: no message from ScoutServer` if the Database Layer has heard nothing from its Scout Server for 15 minutes. It repeats the warning every hour. A Scout Server without device activity is quiet too, so check the values of `linkState` and `keyMatchesDatabase` in the warning before you act.

To change the 15 minutes, set `ScoutServerSilentThresholdSeconds` in the `[Network]` section of `dblayer.ini` to a number of seconds from 60 to 86400, and then restart the **Scout Board Database Layer** service. For any other value, the Database Layer uses 900 seconds.

`dblayer.log` is a UTF-16 file. To search it in Windows PowerShell, use `Select-String` with `-Encoding Unicode`.

Troubleshooting

If a UniconOS Management Board misses the events of a server, **Node health** shows this server as failing or with a warning, and the hint of the check tells you what to do. This table lists the most common causes and their solutions:

SYMPTOM	CAUSE	SOLUTION
Status changes of devices appear only after you reload the page, or a mirroring session ends with **The mirror**	This UniconOS Management Board misses the events of another server.	Open **Node health** in this UniconOS Management Board, and solve the problem of the failing server.

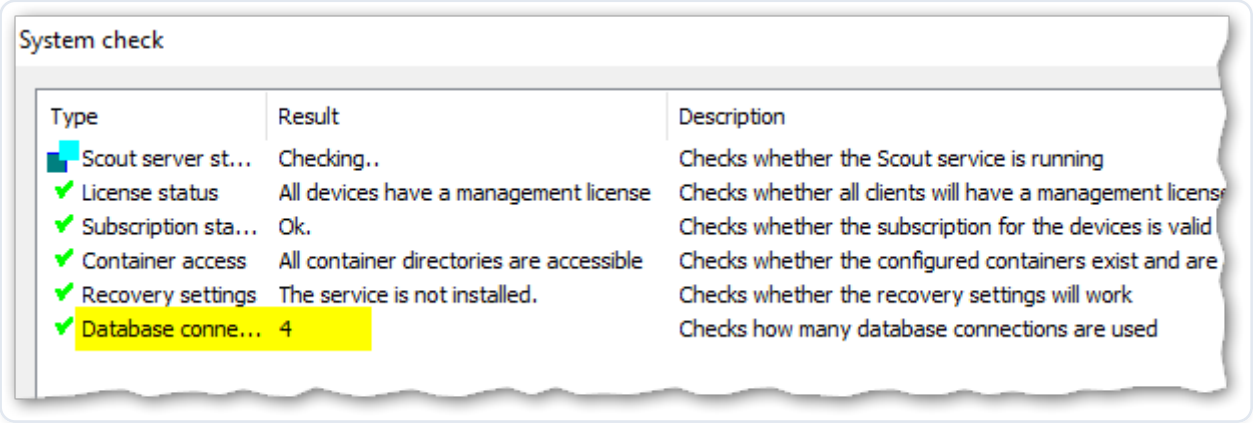
SYMPTOM	CAUSE	SOLUTION
connection could not be set up in time.		
Subscribed by this Scout Board fails, and the hint says that the other server refuses this UniconOS Management Board.	The list of allowed hosts on the other server doesn't contain this server. The `dblayer.log` of the other server shows `Refused a DBLayer client at <address>`.	On the other server, add the IPv4 address of this server to the list, see Adding a Scout Server.
Subscribed by this Scout Board fails, and the hint says that port 22151 must be reachable through the firewall.	A firewall drops the connection, for example a rule that admits only the hosts in the list. The `dblayer.log` of the other server doesn't show the connection. The **Scout Board Database Layer** service of the other server can also have stopped: in the first 2.5 to 4 minutes after it stops, **Node health** shows the same hint.	Check that the **Scout Board Database Layer** service runs on the other server. Open port 22151 there for this server, for example in the firewall rule **Scout DBLayer Publish**. Also check the list of allowed hosts there.
Node health shows another server with **Warning** and the hint **No running database layer is registered for <host>. Check the address or start the DBLayer service.** The summary above the table reads **Some nodes have warnings**.	The **Scout Board Database Layer** service of the other server has stopped or crashed, but its Scout Server runs. This UniconOS Management Board no longer receives the events of the other server, and its `backend.log` shows `ZeroMQ discovery: dropped tcp://<address>:22151 (row removed)`.	Start the **Scout Board Database Layer** service on the other server.

SYMPTOM	CAUSE	SOLUTION
The servers stopped working together after a server got a new IPv4 address.	The lists of allowed hosts and the firewall rules on the other servers still hold the old address.	Update the lists and the firewall rules on all other servers.
Key consistent fails, or `dblayer.log` shows `ScoutServer link key mismatch`.	The Database Layer of this server still uses an old Scout Server key.	Restart the **Scout Board Database Layer** service on this server.
UniconOS Management Board doesn't add a new server, and `backend.log` shows `ZeroMQ discovery: could not resolve <host>`.	`ZERO_MQ_ADDRESS_PUBLISH` names a server that no longer exists.	Remove the entry from the `.env` file of UniconOS Management Board, and restart the **Scout Board** service.
`backend.log` shows `Took the DBLayer CURVE key from`.	`ZERO_MQ_PUBLIC_KEY` in the `.env` file of UniconOS Management Board is missing or wrong.	Copy the value of `DBLayerPublicKey` in the `[Certificates]` section of `dblayer.ini` into `ZERO_MQ_PUBLIC_KEY`, and restart the **Scout Board** service.
A **Scout Board Database Layer** service has stopped, and `dblayer.log` shows `Cannot read ZMQ keys.`	Scout Server created a new key because the Scout database had none, for example after you pointed the installation to a new, empty Scout database. `eluxd.log` shows `event=db_created`.	Start the service again, and restart the **Scout Board Database Layer** service on every other server.
Scout Server and the Scout Board Database Layer service stop right after they start, and the Windows Event Log says that the key for stored secrets can't be loaded.	The cluster keeps its data key in a key file, and this server has no copy of it or a different file.	Copy the key file of the other servers to this server, and start both services, see Data key for stored passwords.

Number of ODBC connections

The number of ODBC connections between the Scout Server and Scout SQL database is defined dynamically on start-up of the server service. Normally, for each CPU kernel, two ODBC connections are defined and used.

The number of database connections currently used can be viewed by using the system check feature (Scout Console **View > System diagnostics > System check**).



From experience, two ODBC connections for each CPU kernel lead to good results considering

- maximum communication performance between Scout Server and SQL database and
- optimum CPU utilization.

Static versus dynamic ODBC connections

You can specify a fixed number of ODBC connections to meet the particular system requirements of a Scout installation. For this, you must define the following parameter in the `eluxd.ini` configuration file of the Scout Server:

FILE	`%SYSTEMDRIVE%\USERS\PUBLIC\DOCUMENTS\UNICON\SCOUT\SERVER\ELUXD.INI`
Section	[ELUXD]
Parameter	`DatabaseConnections=`
Value	*n (n=1-128)*

> **Note:**

> Increasing the number of database connections manually, can lead to CPU overload.

> For further information on modifying `.ini` files, see Advanced file entries in the **Scout** guide.

Preparing for installation

Scout Server and the Scout Keep Alive service or Scout Statistics Service can be installed on the same machine or on different machines.

Make sure that the operating system is provided with up-to-date patches and the required software is installed. For further information, see System requirements.

Before you start the installation, read the following.

Registering Scout Server in the network

Opening ports

Preparing SQL Server databases

Permissions and certificates

Downloading software

Registering Scout Server in the network

To allow automatic registration of the devices, assign the IP address of the Scout Server via DNS or DHCP.

- DNS: Assign the host name ScoutSrv to the IP address. This is the easiest way.

Or:

- Configure one or more DHCP options. For further information, see DHCP configuration.

> **Notes:**

> - If you want to assign another Scout Server at a later time, use the Scout function **Client relocation**. Do not change the DHCP configuration while the devices are running.

- If you do not use DHCP options for Scout, we recommend that you select the option **Ignore Scout Server DHCP options** in the **Device configuration > Network**.

DHCP configuration

-- optional --

Note:

DHCP options can only be applied to eLux® devices.

A new device booting for the first time can retrieve the following information from a DHCP server:

- IP address or name of the Scout Server (option 222)
- List of Scout Servers (option 224)
- ID of the destination OU on the Scout Server (option 223)

This requires configuring the DHCP server via one of the two following methods.

In method 1 (recommended), you define a new vendor class, set the new options, and then apply the values. Method 2 uses the DHCP Standard Options 222, 223 and 224.

The following instructions are based on the DHCP manager of Windows Server 2012.

Note:

Devices placed by option 223 don't send an OU password. If the destination OU is hidden or protected by a password, Scout Server places such a new device in the default OU, unless the setting **Devices that join a protected or hidden OU without sending its password to Scout Server** is set to **Accept and log**, see Devices in the **UniconOS Management Board** guide.

METHOD 1: DEFINING USER-DEFINED VENDOR CLASS

REQUIRES
DHCP server compliant with RFC 2132, supporting user-defined vendor classes. Otherwise use method 2.

1. Open the DHCP manager.
2. Select the relevant DHCP server, and then click **Action > Define**
3. Click **Add** to create a new class:

OPTION	VALUE
Display name	eLux NG
Description	eLux specific options
Code	ELUXNG
(in ASCII column)	The entry is automatically extended with the related hexadecimal number (45 4C 55 58 4E 47).

4. Click **Action > Set Predefined Options**. Then, in the **Option class** list field, select eLux NG.
5. To define one Scout Server, click **Add** and edit the new option as follows:

OPTION	VALUE
Name	Scout Server
Data type	String
Code	222
Description	Name or IP address of the Scout Server

6. To define more than one Scout Server, click **Add...** and edit the new option as follows:

OPTION	VALUE
Name	Scout Server list

OPTION	VALUE
Data type	String
Code	224
Description	Server names/IP addresses, comma-separated

7. To define a specific OU that you can assign new devices to, click **Add...** and edit the new option as follows:

OPTION	VALUE
Name	Scout OU ID
Data type	Long
Code	223
Description	ID of the destination OU

8. To assign the options, for the relevant DHCP server, select either the **Server Options**, the **Scope options** or the **Reservations**. Then click **Action > Configure Options... > Advanced**.

   In the **Vendor class** list field, select . Select each option defined and enter its value into the **Data entry** field:

OPTION	VALUE
222 Scout Server	Name or IP address of the Scout Server
223 Scout OU ID	ID of the destination OU
224 Scout Server list	<Names or IP addresses of the Scout Servers, separated by comma>

METHOD 2: USING DHCP STANDARD OPTIONS

REQUIRES
The DHCP Standard Options [222], [223] and [224] must be available. Otherwise use Method 1.

1. Open the DHCP manager.
2. Select the relevant DHCP server, and then click **Action > Set Predefined Options...**. In the list field, select .
3. Click **Add...** to create the following Standard Options, as described for Method 1:
   - , String, 222
   - Scout Enterprise server list, String, 224
   - OU ID, Long, 223
4. To assign the options, for the relevant DHCP server, select either the **Server Options**, the **Scope options** or the **Reservations**. Then click **Action > Configure Options... > General**.

   Select each option defined and enter its value into the **Data entry** field:

OPTION	VALUE
222 Scout Server	
223 Scout OU ID	ID of the destination OU
224 Scout Server list	<Names or IP addresses of the Scout Servers, separated by comma>

DISABLING DHCP OPTION 12 AS SOURCE FOR HOST NAMES

If you have configured DHCP option 12 (host name), when connecting new devices, you can have the host names set via DHCP. To obtain the host name not via DHCP but from another source, such as the name template defined in UniconOS Management Board (see Device names in the **UniconOS Management Board** guide), prevent the take-over from DHCP option 12. To do so, use a terminal.ini parameter:

- **File:** `/setup/terminal.ini`
- **Section:** `[Network]`
- **Entry:** `IgnoreDHCPHostname`

- **Value:** `true` (The default value is `false`.)

Assign new devices to specific Scout Servers

If you use more than one Scout Server, you can specify in advance to which of them a device will be assigned during the onboarding process.A filter (regular expression) on the MAC address is used as a criterion for the assignment.

The filter rules are defined in an `.ini` file, which is then transferred to the devices as part of a custom feature package in the image.This way, new devices receive the information to which Scout Server they are to connect, even before the first contact to Scout.

The `.ini` file, for example `scoutmapping.ini`, is a text file that is structured according to the pattern below:

```
[Mapping1]
```

```
 identifier=MAC   pattern=[AB][0-9A-F]
$   scoutsrv=scout1.sampletec-01.com   [Mapping2]   identifier=MAC   patte
rn=[CD][0-9A-F]
$   scoutsrv=scout2.sampletec-01.com   [Mapping3]   identifier=MAC   patte
rn=[EF][0-9A-F]$   scoutsrv=scout3.sampletec-01.com
```

Note the following:

- In Scout, the MAC address is displayed as a 12-digit number without separators (example: 901B0E01CE84)
- The filter must be a regular expression that filters on a substring of the MAC address.
- In a PostInstall script and PreUninstall script of the feature package, the `.ini` file must be referenced, example: `./setup/scoutmapping.ini`

For further details, please contact the Unicon™ support.

Opening ports

Open the following ports in the firewall: (provided the default ports are used)

PORT	TYPE	FROM	TO
1433	TCP	Scout Server	MS SQL Server

PORT	TYPE	FROM	TO
1434	UDP	Scout Server	MS SQL Server (Browser service)
22123	TCP	Scout Server (Scout Management / secure)	eLux endpoints
22125	TCP	Scout Server (Scout Management / TLS)	eLux endpoints
22124	TCP	eLux® endpoints	Scout Keep Alive service on the Scout Server
22150	TCP	UniconOS Management Board, Software Management backend	Scout Board database layer (DBLayer) on the Scout Server
22151	TCP	UniconOS Management Board, and the UniconOS Management Boards of all other Scout Servers that share the Scout database	Scout Board database layer (DBLayer) on the Scout Server
22160	TCP	Web browsers of the administrators	UniconOS Management Board
22170	TCP	UniconOS Management Board	Software Management backend, only if it runs on another computer than UniconOS Management Board
5902	WebRTC (TCP, UDP)	UniconOS Management Board -	eLux endpoints

PORT	TYPE	FROM	TO
		web browser (Mirroring eLux desktop)	
80/443	TCP	Clients (HTTP/ HTTPS)	Web server
80/443	TCP	Clients (Firmware-updates via HTTP/ HTTPS)	Web server

Open ports 22150 and 22151 only for the hosts that need them. On a single server, no other computer needs to reach these ports. If several Scout Servers share one Scout database, open port 22151 on every server for all other servers, see Connections between the servers. From UniconOS Management 2611, the installer turns on the client check of DBLayer: DBLayer then accepts only the hosts in its list, and the firewall rules that the installer adds for these two ports admit only those hosts. For more information, see Installing Scout Enterprise Management Suite.

Note that, after the connection has been established, MS SQL Server assigns port numbers between 1024 and 5000 dynamically to its clients and that communication from 1433 to *ANY* must be allowed.

If you create the firewall rule for the Scout Keep Alive service yourself, create it for port 22124, not for the program `ScoutKeepAlive.exe`. A rule for the program doesn't work, because the service receives the keep-alive messages through Windows. From UniconOS Management 2611, the rule that setup creates opens the port, and an upgrade corrects the rule of an earlier version.

> **Note:**

> For detailed information on ports, protocols and network requirements, see Network requirements guide.

> The firewall service must be started.

Preparing SQL Server databases

Make sure that the following requirements are met:

- The Scout Server machines must be provided with suitable ODBC drivers to connect to the SQL database. For further information, see System requirements.
- We recommend that you create the required databases in Microsoft SQL Server before you install the Scout Enterprise Management Suite®. Alternatively, new Scout databases can be created in Microsoft SQL Server during the installation, relevant rights provided.

  > **Note:**
  >
  > For all databases, the database tables are created automatically by the Scout installer. The installer creates them only in an empty database, and it never deletes tables. With Windows authentication, the installer creates the tables as the Scout service account. This account doesn't have to be a local administrator, but it needs the database role **db_owner** during the installation.
  >
  > For further information, see Database support.

- SQL or AD users (**SQL serverInstance> / Security / Logins**) with relevant permissions for all databases

  For further information, see SQL Server users and application roles.

- The Browser service on the SQL server must be started.

Permissions and certificates

Permissions for the Scout Enterprise Management Suite® installation

- AD administrator account, member of the local administrators group on the target system

For information on how to authenticate to SQL Server, see Authentication to SQL Server.

> **Note:**
>
> During the installation, you specify the initial Scout user. This user can sign in to UniconOS Management Board.

Permissions for the web server

On the web server, one or more eLux® containers provide eLux software packages and images.

- Web Server (IIS) role or relevant permissions for the web server used
- Administrator rights on the web server computer, to publish the container folders of **Software**, see Publishing a container to devices

Permissions for UniconOS Management Board

You sign in to UniconOS Management Board with Active Directory or with an identity provider, Microsoft Entra ID or Okta. UniconOS Management Board applies the administrator rights that you manage under **Security & permissions** > **Manage administrators**, see Administrator rights in UniconOS Management Board.

Administrators who create or change images in the **Software** section need the Software rights, see Software rights.

SSL certificates for Scout Keep Alive service and UniconOS Management Board

The communication between eLux and the Scout Keep Alive service. For this, during installation, you are asked to specify a valid SSL certificate. Alternatively, you will be offered a self-signed certificate. For further information, see Certificate for Scout Keep Alive service.

For a secure connection to the UniconOS Management Board interface, also a valid SSL certificate is required. For further information, see Installing UniconOS Management Board.

eLux certificates for software packages

Each container has its own certificates. With them, Scout checks the signatures of the packages that you add to the container. A new container already has the default certificates, which confirm the signatures of the UniconOS software packages. For packages from other signers, for example packages that you build yourself, import their certificates in the **Software** section of UniconOS Management Board, see Certificates.

Downloading software

Before you start the installation, download the `.zip` files required for the software you want to install:

- Scout Enterprise Management Suite®
- eLux® software packages for the relevant operating system version
- USB stick image for the recovery installation of individual devices for the relevant operating system version
- …

1. Sign in to our myelux.com portal.

2. On the Download menu, choose the relevant software:

OPTION	DESCRIPTION / OPTION	DOWNLOAD
eLux	eLux Software Packages Latest operating system versions, LTSR (Long Term Service Release) or CR(Current Release)	Bundle: With `eLuxRP-x.x.x` AllPackages `-x`, you can add all software packages to a container in one step. Import the archive in the **Software library** of UniconOS Management Board, see Adding packages. Individual packages: Under **Release Packages**, all available software packages of an eLux version are provided for download.
eLux	eLux USB Stick Images Ready-to-use images for USB installation of the latest eLux version	< eLux CR version> Recovery Stick < eLux LTSR version> Recovery Stick includes the Citrix Workspace™ app and the VMware Horizon client to connect against a backend
eLux	eLux Portable eLux on a USB stick, based on latest eLux RP version	eLux Portable (no installation needed)
Scout	Scout Enterprise Management Suite Latest LTSR and CR versions	< Scout CR version>* < Scout LTSR version> includes the installation file `ScoutInstaller.exe`
	Scout Agent for Windows for managing Windows devices	Latest version
Scout Cloud Gateway	Gateway to connect your eLux devices through the Internet (latest version)	Virtual Machine template (`.ova`) or Debian package

OPTION	DESCRIPTION / OPTION	DOWNLOAD
Tools	StickWizz	Latest version (StickWizz is also included in the eLux USB Stick images)
	eLux Builder Kit (eLux SDK VM) Development environment	Please contact sales(at)unicon.com
	Win2eLux Migrate from Windows to eLux	Win2eLux for 64-bit Windows Win2eLux for 32-bit Windows

3. To download the relevant file, click the file name or version number. *The software is downloaded in the form of* `.zip` *files.*
4. Unpack the `.zip` files. Don't unpack the AllPackages archive of UniconOS: you import it in UniconOS Management Board as it is.
5. Provide the installation files such as `ScoutInstaller.exe` on a local hard drive.
6. To create a recovery stick, connect an empty USB stick to a USB port. Start the `StickWizz.exe` application of the zip archive, and then write the image to the stick. For further information, see Creating a USB recovery stick in the **eLux recovery procedures** guide.

Installation: Scout Enterprise Management Suite®

The Scout Enterprise Management Suite includes all features that are necessary or useful for managing a client infrastructure, in particular Scout Server and UniconOS Management Board. From UniconOS Management 2611 LTSR, UniconOS Management Board is the only management interface. To build individual firmware images, use the image management that is integrated in Scout Board, see Software. It's part of the standard installation.

> **Note:**
>
> Scout Console, Scout Report Generator, the legacy ELIAS and the Scout Command Interface (`scmd.exe`) are end of life. The installer no longer installs them, and an upgrade to UniconOS Management 2611 removes them. UniconOS Management 2609 is the last release that installs them. For details, see Deprecation.

Features of the Scout Enterprise Management Suite

Installing Scout Enterprise Management Suite

After the initial installation

Unattended installation

Upgrading to newer versions

Modifying Scout Enterprise Management Suite

Uninstalling Scout Enterprise Management Suite

Features of the Scout Enterprise Management Suite®

Scout Enterprise Management Suite is the management solution for cloud devices, Hybrid Clients, mobile devices and PCs running the operating system eLux®. In addition, Windows-based devices can be managed by using basic Scout management features.

From UniconOS Management 2611 LTSR, UniconOS Management Board is the only management interface, and you build your firmware images with the image management that is integrated in UniconOS Management Board. Scout Console, Scout Report Generator, the legacy ELIAS and the Scout Command Interface are end of life and no longer part of the installation. UniconOS Management 2609 is the last release that installs them.

Scout Enterprise Management Suite consists of several components. Most of the components listed below are included in the standard installation, but you can leave them out if you choose the user-defined installation:

COMPONENT	DESCRIPTION	INSTALLATION
Scout Server	The service controls and manages eLux devices as well as Windows devices on which Scout Agent for Windows has been installed.	`ScoutInstaller.exe`
UniconOS Management Board	Web user interface for the management of eLux devices and Windows-based devices	`ScoutInstaller.exe`

COMPONENT	DESCRIPTION	INSTALLATION
	on which Scout Agent for Windows has been installed	
Software Management	Image management that is integrated in UniconOS Management Board. It adds the **Software** section to UniconOS Management Board, where you keep the software packages in containers and build the firmware images for your devices.	`ScoutInstaller.exe`
Software Management backend	Windows service that powers Software Management and stores its containers and images. It runs next to UniconOS Management Board or on a computer of its own.	`ScoutInstaller.exe`
Recovery service	Customized TFTP service to realize a PXE recovery environment for eLux endpoints	`ScoutInstaller.exe`
Scout Keep Alive service (Replaces the Scout Statistics Service from Scout 15 2209)	Windows service processing keep alive packets from eLux devices	`ScoutInstaller.exe`
Web API	Application programming interface for the management of eLux devices and for the management of Windows-based devices on which Scout Agent for Windows has been installed	`ScoutInstaller.exe`

COMPONENT	DESCRIPTION	INSTALLATION
Scout Database Connection Editor	Tool for changing the database connection of Scout Server, the Scout Board database layer and the Scout Keep Alive service	`ScoutInstaller.exe`

Scout Cloud Gateway and Scout Agent for Windows have their own installers:

Scout Cloud Gateway	Cloud gateway with VPN backend for convenient connection of devices from the Internet to a Scout infrastructure	separate
Scout Agent for Windows	Service providing an interface for Windows-based devices to be managed through Scout Enterprise Management Suite	separate

For how to use these components, see their guides:

- UniconOS Management Board
- Scout Cloud Gateway
- Scout Agent for Windows

Recovery procedures for eLux devices are described in the **eLux Recovery procedures** short guide.

Installing Scout Enterprise Management Suite®

> **Note:**

> Before you start to install, make sure you have read the topics System requirements and Preparing for installation.

> When you run the Scout Enterprise Management Suite installer, we recommend the standard installation scope. It includes, for example, Scout Server, UniconOS Management Board and Software Management, the image management that is integrated in UniconOS Management Board. From UniconOS Management 2611 LTSR, the installer no longer offers Scout Console, Scout Report Generator or the legacy ELIAS. These tools are end of life, and UniconOS Management Board is the only management interface.

Alternatively, choose the `user-defined` option and select the features to be installed. For further information on the provided components, see Features of the Scout Enterprise Management Suite.

> **Notes:**

> - Run the installer from a local hard drive. Do not use a USB stick, CD-R drive or a network drive.
> - Anti-virus programs can have an impact on the installation. If required, disable the anti-virus program before you perform the installation.

1. Run the `ScoutInstaller.exe` file. Setup asks for administrator rights. For an unattended installation, run the file in a command prompt that runs as administrator, see Unattended installation.
2. Select the installation language. Subsequently, read and accept the license terms.
3. Select the database type you want to use:

   - Microsoft SQL Server

   For further information, see Database support.
4. On the page **Connect Scout to your License Server**, enter the **License server URL**: the host name of your Citrix® License Server and, if needed, its port. Scout gets its licenses from this License Server. You can configure the License Server later in the UniconOS Management Board UI at any time.

   Scout needs Citrix License Server 2607 (build 56200) or later. If the License Server is older, or if setup can't read its version, setup shows a warning. Click **OK** to install anyway, or **Cancel** to enter another License Server. With an older License Server, Scout can't activate its licenses: it runs on its license grace period and stops serving devices when the grace period ends.

You can also continue without a License Server, or with one that isn't reachable or configured yet. In this case, Scout will go into a grace period, until a valid License Server has been set up, see Operation and grace period.

> **Notes:** To see more information about the Citrix License Server and the licensing process, please follow the following links:

> Citrix License Server

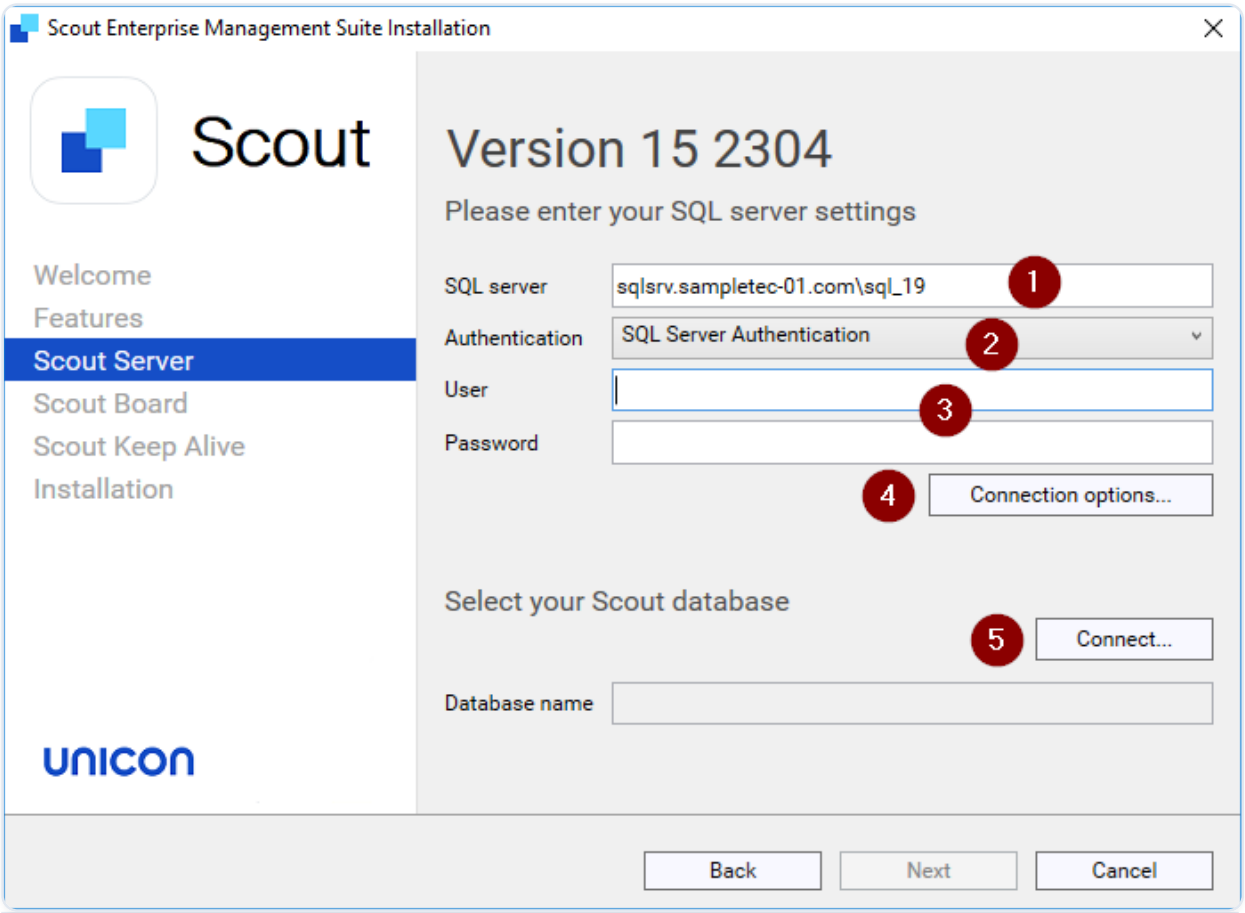
> Citrix License Manager

5. Choose the installation scope. To select individual features for installation or change the installation directory, select `User-defined`. For the standard installation, select `Standard`.
6. Specify the database connection data for your **Scout database**.

> **Notes:**

> Microsoft SQL LocalDB is not supported.

> Specify the required data to connect to your Microsoft SQL Server machine:

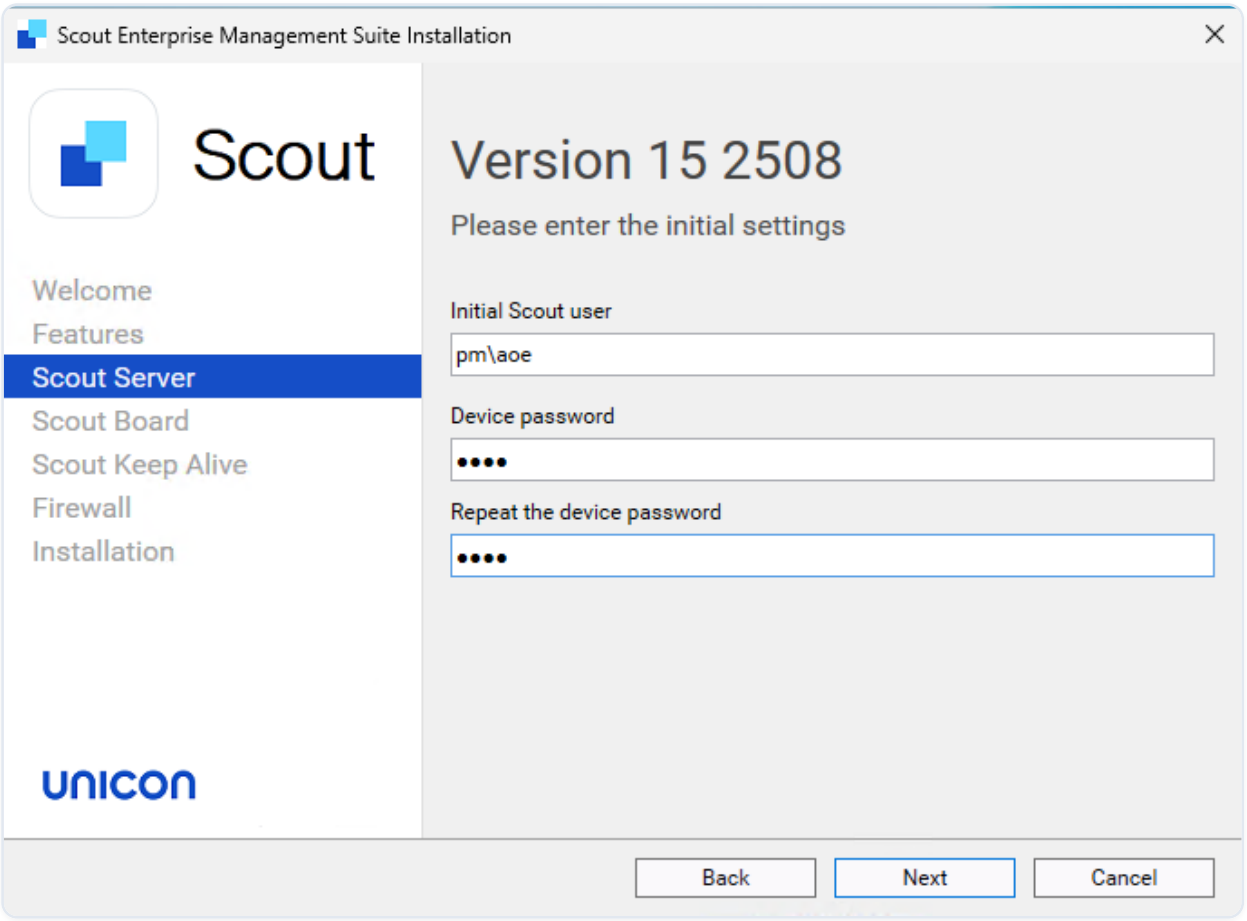


1	*<SQL Server machineinstance>* Example: `sqlsrv.sampletec-01.comsql_19`
2	`SQL Server authentication` or `Windows authentication` For further information, see Authentication to SQL Server.
3	SQL or Windows username and password for database access
4	SQL Server connection options: *For AlwaysOn Cluster: Faster reconnection after fail-over*, Use encrypted ODBC connection, and Trust server certificate (enabled by default)
5	Click to connect to the database server.

Click **Connect...**, and then, from the list-field, select your **Scout** database.

> **Note:**
>
> To show the databases on the specified SQL server, the SQL Server Browser service must be active.
>
> *Next to **Database name**, the selected database will be shown.*
>
> **Note:**
>
> If you choose "Windows Authentication" for the authentication method, all the services that need to connect to the database will be started by the given user and credentials. All other services will be started by the "Local System" user
>
> If setup can't use the database that you selected, it stops and tells you why. The message names the server, the database and the account, and it quotes the error that SQL Server or Windows reported. The database stays unchanged. For example, setup stops if it can't read the Scout version of the database, or if the database contains tables but no Scout version. Setup creates the Scout tables only in an empty database, and it never deletes tables.

7. In the dialog for initial settings, provide the initial admin user and the initial device password for eLux® endpoints. The initial admin user can be used to logon to the UniconOS Management Board UI.



8. In the next dialog, edit the **UniconOS Management Board** options. (optional component, from Scout 15 2209)

   Enter the port number for the Scout Board service and the computer name, see Installing UniconOS Management Board in the **UniconOS Management Board** guide.

9. If you install the Software Management backend, set its port (default `22170`) and its data folder (default `C:\ProgramData\Unicon\Software`) on the page **Please configure the Software Management backend**. The data folder holds your containers and needs enough free disk space, see Installing Software Management.

10. In the next dialog, edit the options for the **Scout Keep Alive service** database. (optional component, from Scout 15 2209)

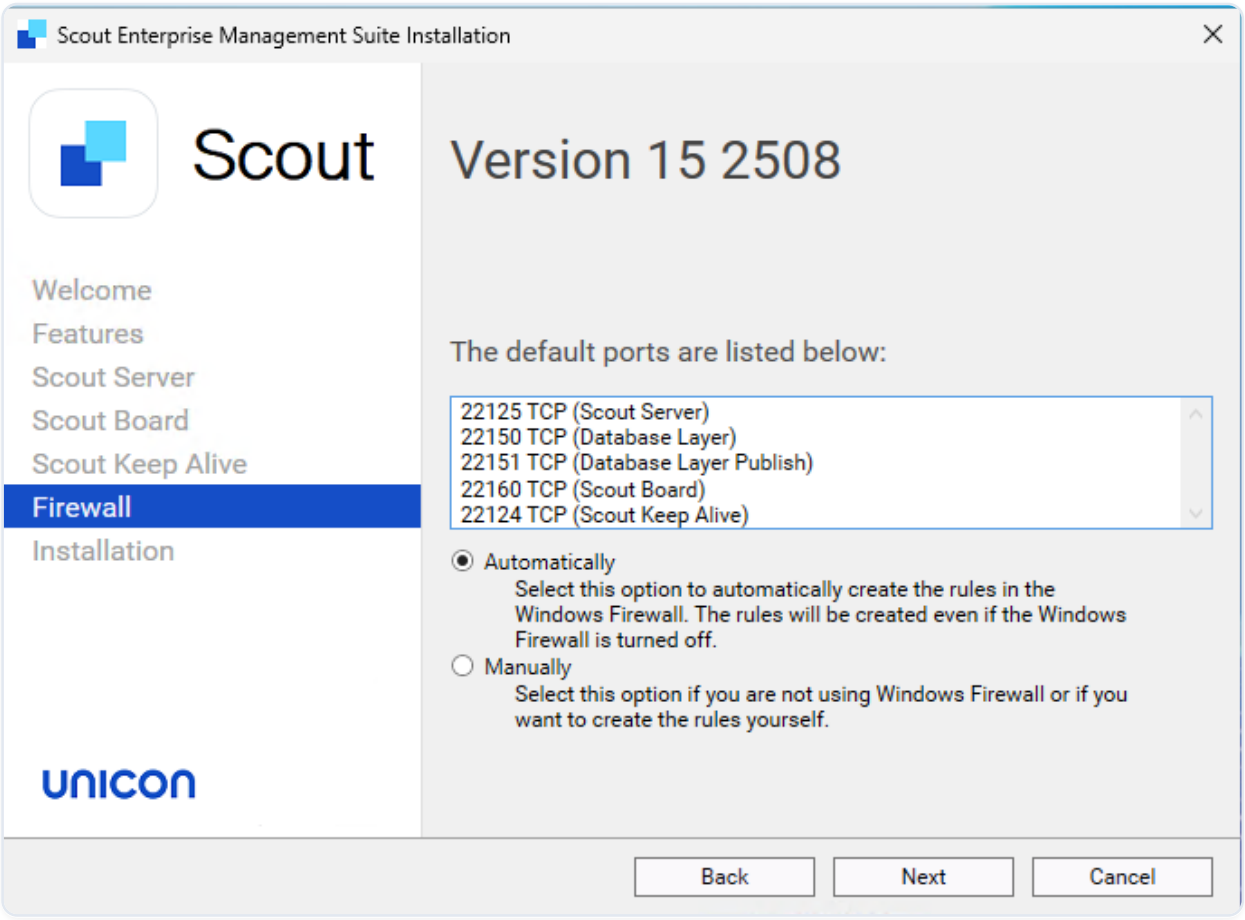
    Specify the TCP port.

    With **HTTPS**, a secure connection to the interface is used. Specify a valid SSL certificate. Alternatively, let the system create a self-signed certificate, and then continue with the Scout installer.

> **Note:**

> Click to update the list-field content and select the newly created certificate. Any certificates available are shown with their assigned friendly names. If there is no friendly name or it is assigned more than once, their serial number is shown.

11. In the firewall dialog, you can see all the ports that need to be activated, in order for Scout, UniconOS Management Board and Scout Services to work properly and for the communication with eLux endpoints. These firewall rules can be added automatically by the installer or you can activate them later manually.



On the same page, the option **Allow only these hosts to connect to DBLayer (ports 22150 and 22151)** is selected for a new installation, with `self` (this computer) in the list. The database layer of UniconOS Management Board (DBLayer) then refuses every host that isn't in the list. Enter the hosts as `self`, IPv4 addresses or IPv4 ranges such as `10.0.0.0/24`, separated by commas. Add every computer that runs a UniconOS Management Board or a Software Management backend for this Scout Server. If several Scout Servers share the Scout database, also add the IPv4 addresses of all other Scout Servers, see Adding a Scout Server. If setup adds the firewall rules automatically, the rules for ports 22150 and 22151 also admit only the listed hosts. For how to configure DBLayer, see Installing UniconOS Management Board.

12. In the last step, check the overview of all features to be installed. To start the installation process, click **Install**.

If required software components such as the Visual C++ Redistributable are not yet available on the target system, they will be installed by the installer.

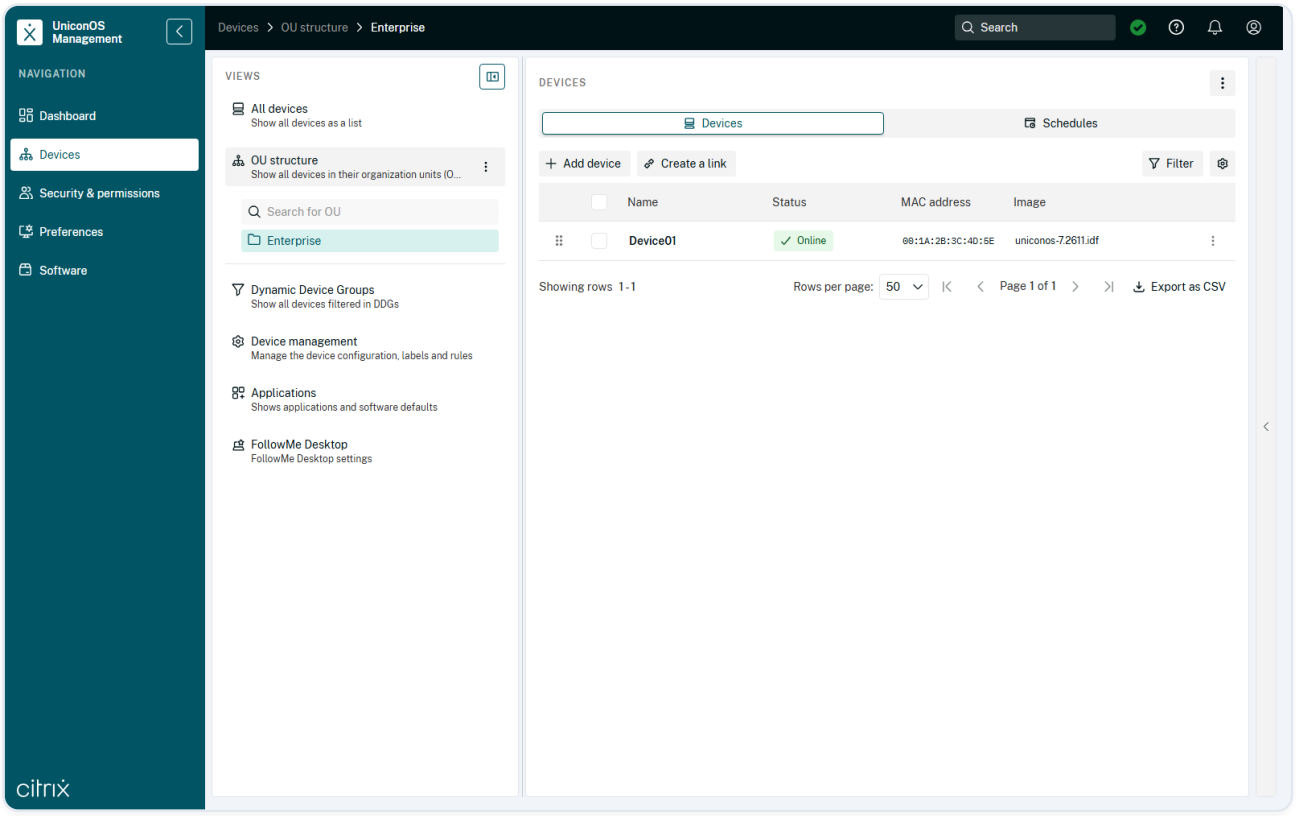
Once the installation is completed, open UniconOS Management Board in your browser, see Logging in. In the Scout group of the Windows Apps view, you will find the Scout Database Connection Editor.

After the initial installation

The newly installed Scout Enterprise Management Suite® gets its licenses from the Citrix® License Server that you entered during the installation. As long as Scout doesn't get a valid license from it, Scout runs on a grace period, see Operation and grace period.

After you have completed the installation with an empty Scout database, you manage your devices in UniconOS Management Board. Sign in with the initial Scout user that you specified during the installation, see Logging in. The new Scout database already contains a few objects that you can start from:

- On the top level, three applications are predefined to connect to a back-end: **RDP**, **VMware Horizon** and the **Firefox** web browser.
    - To use one of the applications, modify the properties of its application definition and provide the relevant software for the devices in their firmware image. To see the application definitions in UniconOS Management Board, click **Devices** in the **Navigation** menu, and then click **Applications** in the **Views** panel.
- A top-level organizational unit (OU) named **Enterprise** is created.
    - Under the top-level OU, add further OUs corresponding to your corporate structure. For further information, see Adding OUs in the **UniconOS Management Board** guide.



Unattended installation

> **Note:**
>
> The following information refers to Scout Enterprise Management Suite® 15.2302 and later. It includes the installation of UniconOS Management Board and Scout Keep Alive service.

Performing an unattended installation of the Scout Enterprise Management Suite

- In a command prompt that runs as administrator, run the `ScoutInstaller.exe` file along with the required command-line parameters: `"ScoutInstaller.exe" /s /v"<parameter>" /v"<parameter>"` You can add as many parameters as you like.

Creating a command line for unattended installation

Create a batch file for unattended installations by using a one-time manual installation according to the desired criteria.

1. Perform a manual installation of the Scout Enterprise Management Suite with the required components and options.
2. Open the log file created during the installation in a text editor: `%LOCALAPPDATA%\Temp\Scout_Enterprise_Management_Suite_<time stamp>.log`

3. Under `Silent install` command line, copy the command line created by the manual installation.
4. Create a batch file that contains the copied command line. *The log never contains passwords, license tokens or secret keys, so you must add them to the command line yourself.*
5. Replace the <`SET_PASSWORD`> strings for the database passwords by the relevant clear text passwords. If you want to use encrypted passwords, add_CRYPTED to the parameter names, see below.

List of parameters

The following tables summarize the available parameters and the values they can be given. On the left, **default values** are displayed in **bold**.

> **Note:**
>
> To encrypt passwords, you can use environment variables. For further information, see Encrypting values.

> **Parameters for /v**:

PARAMETER	DESCRIPTION
UCPROP_DBTYPE=`2`	`2`=Microsoft SQL Server
UCPROP_DBNAME=`Scout`	Name of the Scout database
UCPROP_DBSERVER=`sqlsrv.sampletec-01.comsql_12`	Database server (and instance) of the Scout database
DB_SCOUT_DB_AUTHENTICATION=`Windows Authentication`	Windows Authentication SQL Server Authentication
UCPROP_DBUSER=`Scout-Admin`	Only for SQL Server authentication:SQL username for the Scout database
UCPROP_DBPASSWORD_CRYPTED=`u[D Gqu[w_`	Only for SQL Server authentication:Crypted password for the Scout database, see eluxd.ini

PARAMETER	DESCRIPTION
UCPROP_DBPASSWORD=`My_Password`	Only for SQL Server authentication:Unencrypted password for the Scout database
UCPROP_SERVICEUSER	Only for Windows authentication:Windows username
UCPROP_SERVICEPASSWORD_CRYPTED	Only for Windows authentication: Encrypted Windows password
UCPROP_TRUSTSERVERCERTIFICATE=`1`	Trust the database server certificate
UCPROP_ENCRYPT=`0`	`0`: Use unencrypted ODBC connection, `1`: Use encrypted ODBC connection
UCPROP_MULTISUBNETFAILOVER=`0`	`0`: Default, `1`: Faster reconnection after fail-over (AlwaysOn Cluster)
UCPROP_DBCREATE=`0`	`0`: Setup uses the existing Scout database, `1`: Setup creates the Scout database on the database server. Setup never deletes a database or its tables.
UCPROP_LANGUAGE=`1`	Display language: `0`: German, `1`: English (If the parameter is not set, the language defined in the operating system is used.)
RUNSCOUTSERVICE=`true`	`true`: Scout services will be started during installation, `false`: Scout services will not be started

In a silent upgrade, the service account and database credentials that you pass with `/v` take precedence over the ones that setup saved. Without them, setup uses the saved credentials. Setup doesn't save a new service account password in `install.ini`, so after a password change, pass the service account credentials with every silent upgrade.

With Windows authentication, setup uses the service account password from `install.ini` only if `install.ini` names the account that the Scout Server service runs as. Otherwise, for example after you changed the account of the service in Windows, an update and a change of the Database Layer client check run only silently, with `/v"UCPROP_SERVICEUSER=<account>"` and `/`

`v"UCPROP_SERVICEPASSWORD=<password>"`. The setup wizard stops with a message in this case. Apart from the service credentials, an update, a modification and a repair don't take their settings from `install.ini`, but from the installed components.

An update keeps the values of `Encrypt`, `TrustServerCertificate` and `MultiSubnetFailover` that you set in `eluxd.ini`, and it writes them to `dblayer.ini` as well. For Scout Keep Alive, a silent update keeps the values in `scoutkeepalive.ini`, and a wizard update writes the values from `eluxd.ini` there. So set your own values in `eluxd.ini`.

DEFAULT SETTINGS FOR THE DEVICES (BASE DEVICE CONFIGURATION)

UCPROP_DESKTOP_LANGUAGE=`en_US`	Display language for the desktop
UCPROP_KEYBOARD_LANGUAGE=`en`	Keyboard language
UCPROP_TIMEZONE=`US/Eastern`	Time zone

LICENSE SERVER

Scout gets its licenses from a Citrix® License Server, or from an on-premises license file if you have one. Set the license source with these parameters:

LSURL=`my.license.server`	Citrix License Server URL (the port can be added optionally `:port`)
LS=`1`	`1`: Scout uses the Citrix License Server in `LSURL`, `0`: Scout uses the on-premises license file in `UCPROP_ONPREM_LICENSE_TOKEN`
UCPROP_ONPREM_LICENSE_TOKEN=`<token>`	Only with `LS=0`: license token of your on-premises license file. Pass it with `/v`, also when you start the installer with its wizard. A token without `/v` is ignored.

Scout needs Citrix License Server 2607 (build 56200) or later. If the License Server is older, or if setup can't read its version, a silent installation writes a warning to the log file and continues. Scout then runs on its license grace period and stops serving devices when the grace period ends.

In a silent new installation, `LS` and the other values that you pass with `/v` take precedence over the values in an `install.ini` or `eluxd.ini` file that is already on the computer. A silent upgrade keeps the license mode of the installed version and only logs an `LS` value that contradicts it.

The operating modes Enterprise Subscription and Managed Service Provider are no longer available. A silent installation that still passes myelux.com credentials for them stops before it installs anything and writes the reason to the log file.

FEATURES

ADDLOCAL=`CommonFeature,Server,Recovery,ImageManagement`	Only the specified Scout features will be installed: `CommonFeature`, `Server`, `Recovery` and `ImageManagement` (Software Management, the image management in UniconOS Management Board)
INSTALL_SCOUT_FEATURE=`1`	`0`: Scout components will be skipped, `1`: Scout components will be installed (as defined in ADDLOCAL)
INSTALL_SCOUTBOARD_FEATURE=`1`	`0`: UniconOS Management Board will be skipped, `1`: UniconOS Management Board will be installed
INSTALL_SCOUTKEEPALIVE_FEATURE=`1`	`0`: Scout Keep Alive service will be skipped, `1`: Scout Keep Alive service will be installed

UNICONOS MANAGEMENT BOARD

UCPROP_SCOUTBOARD_HOST=`scout.sampletec-01.com`	Computer name (FQDN) of the machine on which you want the database layer to run
UCPROP_SCOUTBOARD_PORT=`22160`	Port number of the Scout Board service
UCPROP_SCOUTBOARD_DBLAYER_ADDRESS=`tcp://scout.sampletec-01.com:22150`	Scout Board database layer address

UCPROP_SCOUTBOARD_DBLAYER_ADDRESS_PUBLISH=`scout.sampletec-01.com:22151`	Public address of the Scout Board database layer
UCPROP_SCOUTBOARD_AUTH_TYPE=`AD, OKTA_OIDC, OKTA_SAML, ENTRAID_OIDC, ENTRAID_SAML`	Preferred authentication type
UCPROP_SCOUTBOARD_IDP_ISSUER=`your-issuer-url`	OIDC issuer URL
UCPROP_SCOUTBOARD_IDP_CLIENT_ID=`your-client-id`	OIDC client id
UCPROP_SCOUTBOARD_IDP_CLIENT_SECRET=`your-client-secret`	OIDC client secret
UCPROP_SCOUTBOARD_IDP_INITIAL_USER=`your initial IdP user`	The initial IdP user
UCPROP_SCOUTBOARD_IDP_SSO_ENTRY=`your-sso-entry-url`	SAML SSO entry URL
UCPROP_SCOUTBOARD_IDP_CERTIFICATE_PATH=`your-certificate-path`	SAML certificate path
UCPROP_SCOUTBOARD_IDP_PRIVATE_KEY_PATH=`your-provate-key-path`	SAML key path
UCPROP_SCOUTBOARD_IDP_LOGOUT_URL=`your-logout-url`	SAML logout URL

SOFTWARE MANAGEMENT BACKEND

The Software Management backend stores the containers and builds the images for the **Software** section of Scout Board. These parameters install and configure it:

PARAMETER	DESCRIPTION
INSTALL_ELIASBACKEND_FEATURE=`1`	`1`: the Software Management backend is installed (default on a new installation), `0`: it is skipped. Software Management in

PARAMETER	DESCRIPTION
	UniconOS Management Board also needs `ImageManagement` in `ADDLOCAL`.
UCPROP_ELIAS_BACKEND_PORT=`22170`	Port of the Software Management backend
UCPROP_ELIAS_APP_ROOT_DIR=`C:\ProgramData\Unicon\Software`	Data folder of the Software Management backend. It holds the containers.
UCPROP_ELIAS_TLS_CERT_FILE, UCPROP_ELIAS_TLS_KEY_FILE	Certificate and private key of the backend as PEM files. Without a certificate, setup creates a self-signed one.
UCPROP_ELIAS_TLS_PFX_FILE, UCPROP_ELIAS_TLS_PFX_PASSWORD	Certificate of the backend as a PFX file with its password, instead of the PEM files
UCPROP_ELIAS_BACKEND_URL=`https://localhost:22170`	Address at which UniconOS Management Board reaches the backend. Change it when the backend runs on another computer.
UCPROP_ELIAS_TLS_CLIENT_CA_FILE	Required for a backend without UniconOS Management Board on the same computer: path of a copy of the file `elias-client\ca.crt` of UniconOS Management Board. An upgrade keeps the installed file.
UCPROP_ELIAS_DBLAYER_PUBLIC_KEY	Only for a backend without UniconOS Management Board on the same computer: the key of the Scout Board database layer, `ZERO_MQ_PUBLIC_KEY` in the `.env` file of UniconOS Management Board
UCPROP_ELIAS_SCOUT_BOARD_SESSION_KEYS	Only for a backend without UniconOS Management Board on the same computer: the registry value `SessionPublicKeys` under `HKLM\SOFTWARE\Unicon\Scout\ScoutBoard` on the UniconOS Management Board computer. An upgrade or repair keeps the installed value.
UCPROP_ELIAS_SCOUT_BOARD_GRANT_KEYS	Only for a backend without UniconOS Management Board on the same computer: the registry value `GrantPublicKeys` under

PARAMETER	DESCRIPTION
	`HKLM\SOFTWARE\Unicon\Scout\ScoutBoard` on the UniconOS Management Board computer. An upgrade or repair keeps the installed value.
UCPROP_FIREWALL_ADD_RULES_ELIASBACKEND=`0`	`1`: adds a firewall rule for the port of the backend. Only for a backend without UniconOS Management Board on the same computer.
ELIAS_CERTIFICATE_KEY_SECRET	Only when you install the backend again after an uninstallation: the registry value `CertificateKeySecret` under `HKLM\SOFTWARE\Unicon\ScoutEliasBackend\Secrets` that you noted before the uninstallation. The backend can then use your signing certificates again, see Uninstalling the backend.
INSTALLDIR_ELIASBACKEND	Install folder of the backend

To install only the backend, on a computer without UniconOS Management Board, skip the other components with `INSTALL_SCOUT_FEATURE=0`, `INSTALL_SCOUTBOARD_FEATURE=0` and `INSTALL_SCOUTKEEPALIVE_FEATURE=0`. Pass the address of the Scout Board database layer on the Scout Server in `UCPROP_SCOUTBOARD_DBLAYER_ADDRESS`, for example `tcp://scout.sampletec-01.com:22150`, together with the parameters above that are marked for a backend without Scout Board. Without the address, setup stops with exit code `1`, and the log file names the missing parameter. Then register the backend on the Scout Server, see Software Management backend on another computer.

FIREWALL SETTINGS

UCPROP_FIREWALL_ADD_RULES_SCOUTBOARD=`1`	`1`: automatically add the firewall rules; `0`: don't add them
UCPROP_FIREWALL_ADD_RULES_SCOUTKEEPALIVE=`1`	`1`: automatically add the firewall rules; `0`: don't add them

UCPROP_FIREWALL_ADD_RULES_SCOUTSERVER=`1`	`1`: automatically add the firewall rules; `0`: don't add them
UCPROP_FIREWALL_SCOUTBOARD_PORT=`22160`	Firewall rule to add
UCPROP_FIREWALL_DBLAYER_PORT=`22150`	Firewall rule to add
UCPROP_FIREWALL_DBLAYER_PUBLISH_PORT=`22151`	Firewall rule to add
UCPROP_FIREWALL_SCOUTKEEPALIVE_PORT=`22124`	Firewall rule to add
UCPROP_FIREWALL_SCOUTSERVER_PORT=`22125`	Firewall rule to add

A silent installation has no default ports. To let it create firewall rules, pass the switch together with all port numbers of its rules, for example:

```
/v"UCPROP_FIREWALL_ADD_RULES_SCOUTSERVER=1" /
v"UCPROP_FIREWALL_SCOUTSERVER_PORT=22125"

/v"UCPROP_FIREWALL_DBLAYER_PORT=22150" /
v"UCPROP_FIREWALL_DBLAYER_PUBLISH_PORT=22151"
```

DATABASE LAYER CLIENT CHECK

The database layer of UniconOS Management Board (DBLayer) can check which hosts connect to it. With the client check set to `Enforce`, DBLayer accepts connections only from the hosts in its list. With these parameters, you set the client check and the hosts that may connect:

PARAMETER	DESCRIPTION
DBLAYERCLIENTCHECK=`Enforce`	`Enforce`, `Audit` or `Off`. If you don't set the parameter, a new installation uses `Enforce`. An upgrade keeps an existing setting; without one, it also uses `Enforce`.

PARAMETER	DESCRIPTION
DBLAYERALLOWEDCLIENTS=`self`	Hosts that may connect to DBLayer: `self` (this computer), IPv4 addresses and IPv4 ranges such as `10.0.0.0/24`, separated by commas. Add every computer that runs a UniconOS Management Board or a Software Management backend for this Scout Server.
DBLAYERSCOUTBOARDKEYS=`<key>`	Only on the DBLayer server: registers the key of a UniconOS Management Board on another computer. The last setup page of that UniconOS Management Board shows the key.
DBLAYERELIASKEYS=`<key>`	Only on the DBLayer server: registers the key of a Software Management backend on another computer. The last setup page of that backend shows the command line with the key.

Setup checks these values before it installs anything. With `DBLAYERCLIENTCHECK=Enforce` and no list, or with a host name or an invalid range in the list, a silent installation stops with exit code 1 and writes the reason to the log file.

With `Enforce`, the firewall rules that setup adds for ports 22150 and 22151 admit only the listed hosts. For what the modes do and how to change the settings later, see Installing UniconOS Management Board.

SCOUT KEEP ALIVE SERVICE

UCPROP_SCOUTKEEPALIVE_DBSERVER=`sqlsrv.sampletec-01.com`	Database server (and instance) for the keep alive data. These are normally stored in the Scout database, but can also be stored in a separate database.
UCPROP_SCOUTKEEPALIVE_DBUSER	
UCPROP_SCOUTKEEPALIVE_DBNAME=`Scout`	
UCPROP_SCOUTKEEPALIVE_DBPASSWORD	

UCPROP_SCOUTKEEPALIVE_AUTHENTICATION=`Windows`	`Windows`, `SQL Server`
UCPROP_SCOUTKEEPALIVE_SERVICEUSER	
UCPROP_SCOUTKEEPALIVE_SERVICEPASSWORD	
UCPROP_SCOUTKEEPALIVE_TRUSTSERVERCERTIFICATE=`1`	
UCPROP_SCOUTKEEPALIVE_ENCRYPT=`0`	
UCPROP_SCOUTKEEPALIVE_MULTISUBNETFAILOVER=`0`	
UCPROP_SCOUTKEEPALIVE_CERTIFICATES=`CREATESELFSIGN`	Certificate name or automatically created self-signed certificate
UCPROP_SCOUTKEEPALIVE_PORT=`22124`	Port number of the Scout Keep Alive service

FURTHER PARAMETERS

`/s`	The installation will be performed in unattended (silent) mode.
`/uninstall`	The Scout Enterprise Management Suite will be uninstalled.
`/l "%PUBLIC%DocumentsUniConscoutlog.txt`	The log file will be forwarded to the specified file.

Example for an unattended installation

```
ScoutInstaller.exe /s /v"UCPROP_DBTYPE=2" /v"UCPROP_DBNAME=Scout"

/v"UCPROP_DBSERVER=sqlsrv.sampletec-01.comsql_12"

/v"DB_SCOUT_DB_AUTHENTICATION=SQL Server Authentication"
```

```
/v"UCPROP_DBUSER=Scout-Admin"

/v"UCPROP_DBPASSWORD_CRYPTED=u[D``Gqu[w_" /v"UCPROP_DESKTOP_LANGUAGE=de"

/v"INSTALL_SCOUTBOARD_FEATURE=1"

/v"UCPROP_SCOUTBOARD_HOST=scout.sampletec-01.com"

/v"UCPROP_SCOUTBOARD_PORT=22160"

/v"UCPROP_SCOUTBOARD_DBLAYER_ADDRESS=tcp://scout.sampletec-01.com:22150"

/v"UCPROP_SCOUTBOARD_DBLAYER_ADDRESS_PUBLISH=tcp://
scout.sampletec-01.com:22151"

/v"ADDLOCAL=CommonFeature,Server,Recovery,ImageManagement"

/v"INSTALL_SCOUTKEEPALIVE_FEATURE=0"

/v"INSTALLDIR=C:/Program Files/Unicon/Scout"
```

Note:

When you perform an attended installation and select the required options, an eluxd.ini file will be created in the Scout Server directory. This file contains several Scout values that might be useful for unattended installation.

Exit codes

A silent installation reports its result with one of these exit codes:

EXIT CODE	MEANING
`0`	The installation completed.
`1`	The installation failed. The log file names the reason.
`3010`	The installation completed, and Windows must be restarted. Restart the computer before you

EXIT CODE	MEANING
	run the installer again. Earlier versions returned `0` in this case.

Performing unattended uninstall

- Run the following command: `ScoutInstaller.exe" /s /uninstall`

Upgrading to newer versions

An existing Scout Enterprise Management Suite® installation can be updated to the latest version in just a few steps.

1. Perform a full database backup of your Scout databases.
2. Download the latest version of the Scout Enterprise Management Suite as a `.zip` file from our technical myelux.com
3. Unpack the `.zip` file and provide the installation file on a local hard disk.
4. Run the `ScoutInstaller.exe` file. Setup asks for administrator rights before it changes the installation. For a silent update, run the file as administrator.
5. Follow the instructions of the installation Wizard. Select your existing Scout databases.

Depending on the extent of new features, upgrading to a new version might cause longer runtimes when the Scout database is converted.

Setup needs the Scout database during the upgrade. If it can't connect to the database, it stops with a message that names the reason, and it rolls the upgrade back.

If several Scout Servers share one Scout database, upgrade them one after the other. For the upgrade to UniconOS Management 2611, keep to the order in Several Scout Servers on one database.

> **Important:**

> To upgrade to Scout 15 2204 or later versions, the upgrade base must be Scout 15.2.0 or later. Any older databases cannot be converted and will not be accepted by the installer.

Upgrading to UniconOS Management 2611

An upgrade to UniconOS Management 2611 LTSR removes the legacy tools, checks the version of your License Server, turns on the client check of the Scout Board database layer and converts stored passwords to a new format. Scout Server also checks the access passwords of OUs itself. Read this section before you start the upgrade.

REMOVED LEGACY TOOLS

From UniconOS Management 2611 LTSR, the installation no longer includes Scout Console, Scout Report Generator, the legacy ELIAS and the Scout Command Interface (`scmd.exe`). These tools are end of life, and UniconOS Management 2609 is the last release that installs them. An upgrade to 2611 removes them:

- Before the upgrade starts, the installer shows a notice that names the tools that the upgrade removes and what replaces them, for example **This update removes Scout Console, Report Generator, ELIAS and scmd.exe. UniconOS Management Board and its REST API replace them.** A silent upgrade writes the same text to the log file.
- The upgrade deletes the program files and shortcuts of these tools. Files that you saved in `%PUBLIC%\Documents\Unicon\Elias` and `%PUBLIC%\Documents\Unicon\Scout\Report` stay.
- The Microsoft Report Viewer runtime that earlier versions installed for Scout Report Generator stays on the computer.
- The upgrade deletes `scmd.exe` from the Scout Server folder. If you put a copy back into this folder by hand, the next repair or update deletes it again. To automate management tasks, use the UniconOS Management Board REST API instead, see Using the REST API.
- Remove Scout Console from every other computer before you upgrade. The upgrade removes the tools only on the computer where it runs, and an earlier Scout Console on another computer can still connect to the upgraded Scout database after a warning about its version. Scout Console isn't tested with UniconOS Management 2611.
- On a computer that has only these tools, without Scout Server, Recovery service, UniconOS Management Board or Scout Keep Alive service, the upgrade leaves no Scout installation. Programs and Features then shows no Scout entry. The Windows Firewall rules that setup created for Scout stay. Delete them if you don't need them.

From 2611, UniconOS Management Board is the only management interface, and you build your images with Software Management in Scout Board. An upgrade adds Software Management only if you select it. On the page **Select the Scout features for installation**, the upgrade selects only the components that are already installed. To add Software Management, select **Software**

Management and **Software Management backend** there as well, and **UniconOS Management Board** if your installation doesn't include it yet, because Software Management needs UniconOS Management Board on the same computer. You can also add these components later: run setup again, choose **Modify**, and select them, see Installing additional features or uninstalling unneeded features. For a backend on another computer, see Software Management backend on another computer.

LICENSE SERVER VERSION

Scout needs Citrix® License Server 2607 (build 56200) or later. If Scout uses a License Server, setup checks its version before the upgrade. If the License Server is older, or if setup can't read its version, setup shows a warning. Click **OK** to upgrade anyway, or **Cancel** to keep the installed version. A silent upgrade writes the warning to the log file and continues.

Setup also warns before an upgrade if the License Server presents a certificate that Windows doesn't trust and that isn't the one Scout accepted (**License Server certificate**). Click **OK** to upgrade anyway and configure the License Server again in UniconOS Management Board afterwards, or **Cancel** to keep the installed version. A silent upgrade writes the warning to the log file and continues.

After an upgrade with an older License Server, Scout can't activate its licenses. It runs on its license grace period until you upgrade the License Server, and it stops serving devices when the grace period ends. To go back to the installed version, you need to restore your database backup.

UPGRADING SCOUT BEFORE THE DEVICES

If Scout uses a Citrix License Server, upgrade Scout to 2611 before you update the devices to UniconOS 2611. Versions from 2508 LTSR to 2609.1 can give the devices an end date of their software maintenance that no longer moves. Devices then refuse firmware that was built more than 12 months after Scout started to use the License Server. The update log of such a device shows `need subscription`, and the update ends with `no credits (250)`. UniconOS Management 2508 LTSR CU2.1 and 2609.2 have the same fix as 2611.

After the upgrade, Scout gives a device a new end date with its next license request: when the device starts, about every 7 days, and when Scout starts an update job for it. Then run a failed update again.

DEVICES THAT JOIN PROTECTED OUS

From 2611, Scout Server checks the access password of an OU itself when a new device joins the OU, or when a device moves itself to it. By default, also after the upgrade, Scout Server doesn't

admit a device that sends no password: a new device starts in the default OU, and a moving device stays in its OU. Devices that are already in an OU aren't affected.

The setup wizard of UniconOS 2611 and earlier doesn't send the password to Scout Server. Devices that get their OU from DHCP options, from a GroupID in their image or from Scout Cloud Gateway don't send it either. If new devices join protected or hidden OUs this way, decide before the upgrade how to place them: with OU filter rules, by moving them after they have joined, or with the setting **Accept and log**, which admits them as before, see Devices.

CLIENT CHECK OF THE DATABASE LAYER

The upgrade turns on the client check of the Scout Board database layer (DBLayer), unless you have configured it already. DBLayer then accepts connections only from the hosts in its list. On the update page, setup shows the selected option **Allow only these hosts to connect to DBLayer (ports 22150 and 22151)** with the hosts that it found: this computer (`self`) and the UniconOS Management Board hosts that it can identify.

- Add every UniconOS Management Board, Software Management backend or other Scout Server of this Scout database that isn't running during the upgrade. DBLayer refuses every host that isn't in the list.
- To leave the client check off, clear the option.
- If a UniconOS Management Board runs on another computer, upgrade it too, and then register it with this Scout Server. The last page of its setup shows its key. After a silent upgrade of that UniconOS Management Board, only its setup log names the key. Until you register it, that UniconOS Management Board can't connect, see Allowing a UniconOS Management Board on another computer.

The upgrade doesn't change the firewall rules. For how to configure DBLayer, see Installing UniconOS Management Board.

SESSION SECRET OF UNICONOS MANAGEMENT BOARD

If you set `JWT_SECRET` for UniconOS Management Board yourself, for example in `scoutboard\.env.custom`, check it before the upgrade. The secret needs at least 32 random characters. With a shorter or repetitive secret, setup stops the upgrade before it changes anything, and UniconOS Management Board 2611 doesn't start with such a secret.

If UniconOS Management Board doesn't start because of such a secret, repair UniconOS Management Board, see Repairing installation. The repair replaces the secret with a random one. A `JWT_SECRET` in `.env.custom` wins over it, so remove the secret there first. Instead of the

repair, you can enter at least 32 random characters in `JWT_SECRET` and restart Scout Board. Afterwards, every administrator signs in again.

STORED PASSWORDS

At its first start, Scout Server 2611 converts the stored passwords to a new format that UniconOS Management 2609 and earlier can't read. Make a full backup of the Scout database before the upgrade: to go back to the earlier version, you need to restore it. If several Scout Servers share the database, keep to the order in Several Scout Servers on one database. If you want to keep the data key in a key file, create the file before the upgrade. For further information, see Data key for stored passwords.

SEVERAL SCOUT SERVERS ON ONE DATABASE

If several Scout Servers share one Scout database, upgrade them in the following order. This order makes sure that no Scout Server of an earlier version works with the converted passwords, and that setup finds the other servers for the client check of DBLayer. While Scout Server is stopped on a server, this server serves no devices, so plan a maintenance window.

1. Write down the IPv4 address of every server.
2. Make sure that UniconOS Management Board and the Scout Board Database Layer service run on every server.

   Setup builds the list of allowed hosts from the computers that are connected to DBLayer when the upgrade starts. In version 2609 and earlier, the UniconOS Management Board of another server is connected to this server only if this server is listed in the settings of that UniconOS Management Board. Therefore, setup can miss servers.
3. On every server, stop the **Scout Server** service, for example with the Windows PowerShell command `Stop-Service ScoutServer`. Leave UniconOS Management Board and DBLayer running.

   Don't restart a server before you've upgraded it, because Scout Server starts automatically.
4. Upgrade the servers one after the other:

   - In the setup wizard, check the list under **Allow only these hosts to connect to DBLayer (ports 22150 and 22151)** on the update page. Add every other server that's missing.
   - In a silent upgrade, pass the addresses of all other servers, for example `/v"DBLAYERALLOWEDCLIENTS=self,10.0.0.12,10.0.0.13"`. Otherwise, setup uses only the hosts that it finds.

5. Until every server is upgraded, change the configuration only in the UniconOS Management Board of an upgraded server. Never start a Scout Server of an earlier version with this Scout database again.

You don't need to copy or reset keys. The upgrade keeps the Scout Server key in the Scout database, and each DBLayer reads it from there. If you used to clear `ScoutServerPublicKey` in `dblayer.ini` after a setup run, you no longer need to. If you keep the data key in a key file, put the same file on every server before you upgrade it, see Data key for stored passwords.

After the upgrade, check every server:

- In UniconOS Management Board, **Node health** shows every server with the status **OK**. During the first hour, **Last event received** can show **Unknown**. See Checking the health of the servers.
- `dblayer.log` shows the line `DBLayer client check: Enforce, allowed clients: self, …` with all other servers.
- Unless you passed the list yourself, the Windows Application log contains a warning from setup that names the list.
- If the Windows Firewall still has the rule **Scout DBLayer Maintenance** from an earlier version, delete it. Scout no longer uses its port 22153.

UPGRADING FROM 2508 LTSR CU2

If you run UniconOS Management 2508 LTSR CU2 with Scout Console installed, upgrade directly to 2611. An in-place upgrade to 2607 or 2609 fails with Windows Installer error 2771 ("Upgrade of feature Console has a missing component") and is rolled back, so the installation stays on 2508 LTSR CU2. The upgrade to 2611 removes Scout Console.

Modifying Scout Enterprise Management Suite®

If several Scout Servers share one Scout database, modifying and repairing an installation keep this setup, see Scout Server cluster. To let the database layer of UniconOS Management Board (DBLayer) accept another Scout Server of this Scout database, see Changing the hosts that may connect to DBLayer.

Setup needs the Scout database for a modification or a repair. If it can't connect to the database, it stops with a message that names the reason, and it rolls the change back.

Installing additional features or uninstalling unneeded features

1. Use the control panel (Apps and Features), or run the `ScoutInstaller.exe` setup file. Setup asks for administrator rights before it changes the installation.

2. In the Scout installation dialog, select the **Modify** option and click **Next**.

   The installed features are shown with a checkmark.

3. Select the check boxes of the features you want to install. Clear the check boxes of the features you want to uninstall.

   From UniconOS Management 2611, the list no longer includes Scout Console, Scout Report Generator and the legacy ELIAS. These tools are end of life.

   To add Software Management, select **Software Management** and **Software Management backend**. Software Management needs UniconOS Management Board on the same computer. If you clear both, setup removes the backend, but it keeps the data folder of the backend with your containers.

> **Note:**
>
> If you clear the selection of an installed feature, it will be uninstalled.

Changing the hosts that may connect to DBLayer

From UniconOS Management 2611, DBLayer by default accepts connections on ports 22150 and 22151 only from the hosts in its list. With **Modify**, you can change this list, for example when you add a Scout Server that uses the same Scout database:

1. Use the control panel (Apps and Features), or run the `ScoutInstaller.exe` setup file. Setup asks for administrator rights before it changes the installation.
2. In the Scout installation dialog, select the **Modify** option and click **Next**.
3. Keep the selected features, and click **Next** until the **Firewall** page opens.
4. Under **Allow only these hosts to connect to DBLayer (ports 22150 and 22151)**, edit the list. Keep `self`, and separate the entries with commas, for example `self,10.0.0.12,10.0.0.13`.
5. To limit the firewall rules for ports 22150 and 22151 to the listed hosts as well, select **Automatically**. With **Manually**, setup doesn't change the firewall rules.
6. Click **Next**, and finish the wizard.

Instead of the wizard, you can pass the new list in a silent run from a command prompt that runs as administrator, for example `ScoutInstaller.exe /s /v"DBLAYERALLOWEDCLIENTS=self,10.0.0.12,10.0.0.13"`. The list replaces the existing one.

In both cases, setup writes the list to `dblayer.ini` and restarts the **Scout Board Database Layer** service, so plan a short interruption. Afterwards, `dblayer.log` shows the new list in the line `DBLayer client check: Enforce, allowed clients:`. If the line still shows the old list, restart the **Scout Board Database Layer** service. For the settings behind the list, see Configuring the Database Layer client check.

With Windows authentication, the setup wizard can stop with a message about the service account. Then change the list in a silent run, and pass the service account as well, see Unattended installation.

Repairing installation

1. Use the control panel (Apps and Features), or run the ScoutInstaller.exe setup file. Setup asks for administrator rights before it changes the installation.
2. In the Scout installation dialog, select the Repair option and click Next.

The Scout Enterprise Management Suite is checked for missing files, shortcuts and registry settings and will be repaired if necessary.

A repair keeps your settings: the `.env` file of UniconOS Management Board, the client check settings in `dblayer.ini`, the Scout Server key in the Scout database, and the settings and the certificate of the Software Management backend.

Uninstalling Scout Enterprise Management Suite®

1. Use the control panel (Apps and Features), or run the `ScoutInstaller.exe` setup file. Setup asks for administrator rights before it changes the installation.
2. In the Scout installation dialog, select the **Uninstall** option and click **Next**.

Setup removes the Scout components from this computer. The Scout database stays, and so does the data folder of the Software Management backend with your containers.

> **Note:**

> Uninstalling also removes the secret that protects the private keys of the signing certificates in Software Management. If you install the backend again, signing an image fails, unless you pass the secret that you noted before the uninstallation, or add your signing certificates again. For how to note and pass the secret, see Uninstalling the backend.

Installation: eLux® Container

> **Note:**

> You need the eLux container installation described here only for the legacy ELIAS, which is end of life. The legacy ELIAS is no longer part of the installation from UniconOS Management 2611 LTSR. To build your firmware images, use the image management that is integrated in UniconOS Management Board, see Software in the UniconOS Management Board guide.

> If you use the web-based ELIAS 18, there is no need to install the eLux container. In ELIAS 18, you simply import the relevant software packages or the AllPackages bundle. For further information, see Adding packages in the **UniconOS Management Board** guide.

> The eLux container is a compilation of software packages used to create the firmware (IDF) of a client. The administrator selects a subset of the software package pool to define an IDF (Image definition file) that can be installed on the devices.

The software packages can be provided via the container installation on a web or FTP server (legacy ELIAS) or via the ELIAS 18 web service in a MongoDB (locally or in combination with IIS).

For the main operating system versions , eLux RP 6 / 32-bit and eLux RP 5, dedicated containers are provided. Current eLux RP 6 / 64-bit versions are installed under `...eluxng\UC_RP6_X64`.

Exporting a package

Adding packages

Deleting a package

Installing a container

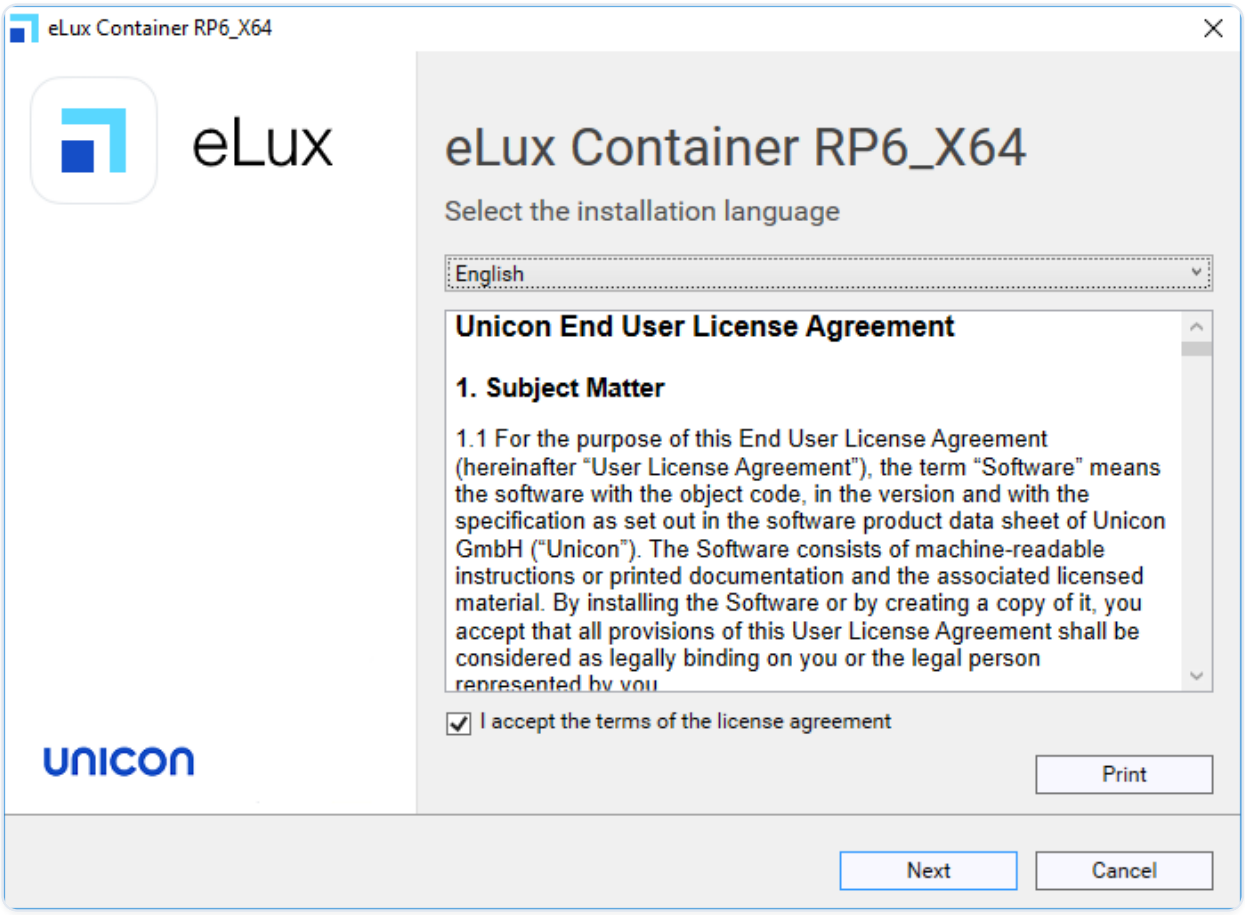
– applies only if you use the legacy ELIAS –

The following instructions show how to install an eLux® container including all software packages that are provided on our portal for the selected operating system version.

> **Note:**

> Before you start to install, make sure you have read the topics System requirements and Preparing for installation.

1. Run the `eLuxContainer.exe` (Extracted Allpackages archive file) as administrator from a local drive.



2. Select the installation language. Subsequently, read and accept the license terms.
3. Select the type of server that you use as the source server for firmware updates:
   - HTTP
   - FTP
4. Specify the access data for the FTP or HTTP server:

OPTION	DESCRIPTION	EXAMPLE
Root directory	Root directory of the server, local or network drive	`W:\inetpub\wwwroot` `C:\Program Files\inetpub\ftproot`

OPTION	DESCRIPTION	EXAMPLE
URL	complete URL to access the server	`http://update.sampletec-01.com` `ftp://update.sampletec-01.com`

5. If you use FTP, in addition, specify the logon data:

OPTION	EXAMPLE
Username	`anonymous`
Password	`elux@sampletec-01.com`

The last dialog summarizes the required hard disk space.

6. To start the installation, click **Install**.

On the web or FTP server, a `UC_RP6_X64` directory is created to hold the eLux RP 6 container. The `container.ini` file and the software packages (`.epm`, `.fpm` and `signature files`) are provided in the container. In ELIAS, you can create your own image now.

7. If the Scout Enterprise Management Suite is already installed, connect from the Scout Console to the eLux container in ELIAS: In the Scout Console, click **Options > ELIAS settings**... and specify the path pointing to the newly created container on the web/FTP server.

Updating to a later version

– applies only if you use the legacy ELIAS –

Updating the eLux® container can become necessary when the latest operating system version or fixes are provided, or when new versions of the client applications are available.

Updating to new major or minor version

New major or minor versions of eLux are provided as releases on our portal.

- Use the latest Allpackages bundle to install a container.

If you upgrade to a new major version, a new container (Example: `UC_RP6_X64`) is created into which the software packages of the new eLux version are installed.

If you update to a new minor version, the new software packages are added to the existing container. The existing software packages remain unchanged.

Updating individual software packages

1. On our portal, from the relevant container, download the required package as a `.zip` file.

   *The **Details** of each package in the container show the relevant enhancements and the history.*

2. Import the `.zip` file into your eLux container by using the ELIAS command **Container > Import Package**. For further information, see Adding packages in the **UniconOS Management Board** guide.

Uninstalling an eLux® container

– applies only if you use the legacy ELIAS –

1. Use the control panel or run the `eLuxContainer.exe` setup file as administrator.
2. In the installation dialog, select the **Uninstall** option and click **Next**.

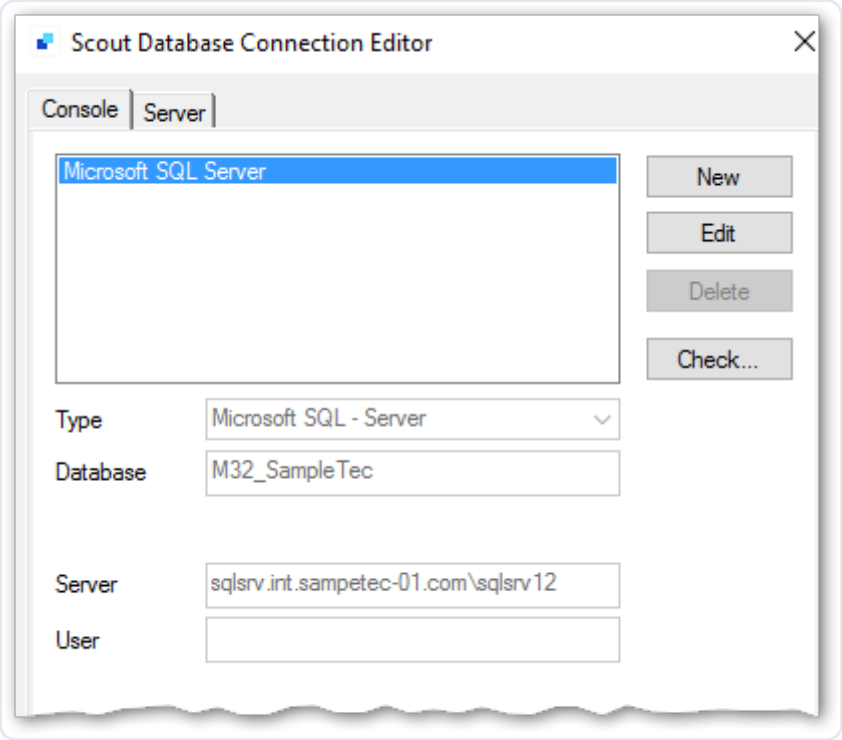
Database connections

When you install the Scout Enterprise Management Suite®, you are asked to specify the Scout database. Check or modify the database connections at any point in time after the installation.

To check or change the database connection of the Scout services, use the **Scout Database Connection Editor**. You can find it as a stand-alone program within the Scout group of the Windows Apps view. On its **Server** tab, you change the database connection of Scout Server, the Scout Board database layer and the Scout Keep Alive service. The **Console** tab applies only to the legacy Scout Console, which is end of life.

Run the Scout Database Connection Editor as administrator: right-click it, and then click **Run as administrator**.

The Scout Database Connection Editor takes its display language from the registry value `Language` under `HKEY_CURRENT_USER\Software\Unicon\Scout\Settings`. If the value doesn't exist, the editor shows German texts. To show English texts, set the value to `1`.



Certificates

Various features and applications require certificates to be provided. For (root) certificates on the client, note the following:

- Unless otherwise stated, the certificates must be Base64-encoded (ASCII) with file name extension `.crt`.
- To transfer certificates to the client, use the Scout feature Files configured for transfer. For further information, see Files configured for transfer in the **Scout** guide.
- On the client, the certificates are stored in the local certificate store `/setup/cacerts/` or in a sub-directory.
- All CA certificates in a trust chain (root and intermediate) must use an RSA key of at least **2048 bits**. eLux validates TLS server certificates with OpenSSL at security level 2, which rejects any chain anchored in a 1024-bit (or weaker) CA with `certificate verify failed` (`verify error: num=67: CA certificate key too weak`). This applies to all OpenSSL-based TLS connections, including Imprivata, firmware update over HTTPS, StoreFront/ICA over HTTPS, and 802.1X/SCEP. Note that Windows and web browsers do **not** enforce this minimum, so a certificate chain that works in a browser can still be rejected on eLux — the browser is not a valid test for this requirement.

The following table provides an overview:

FEATURE	COMPONENT	DIRECTORY
Smart card user logon The certificates are specified in the Scout Console under **Security > User authentication > Certificates**	User authentication / AD+smart card	`/setup/cacerts/login`
Secure connection (TLS)	Firefox	`/setup/cacerts/browser` (Earlier eLux versions have also used `/setup/cacerts/firefox`)
Secure connection (TLS)	Chromium	`/setup/cacerts/browser`
Secure connection (TLS)	Builtin BrowserKiosk mode	`/setup/cacerts/browser`
Secure connection (TLS)	Citrix Workspace-App	`/setup/cacerts/` and `/setup/cacerts/intcerts`
Secure connection (TLS)	VMware Horizon client	`/setup/cacerts/`
Secure connection (TLS)	eLuxRDP	`/setup/cacerts`
Network logon	WLAN drivers / WPA-Supplicant (802.1X)X509/ RadiusNetwork Access Control / SCEP	`/setup/cacerts/setup/cacerts/scep`
VPN / OpenVPN	BaseOS	`/setup/openvpn`
VPN / Cisco AnyConnect	Cisco AnyConnect	`/setup/cacerts` (from eLux RP 6 2302. Earlier versions require the certificate in `/setup/cacerts/ca` and `/setup/cacerts/client`)
Firmware update including certificate check	BaseOS	`/setup/cacerts`

> **Note:**

> StoreFront™ can be called using a Citrix® session or a browser.

Certificate for Scout Keep Alive service

As the eLux® devices and the Scout Keep Alive service communicate via HTTPS, for the installation of the Scout Keep Alive service, a valid SSL certificate for server authentication is required which is bound to port 22124 by default.

As soon as a certificate becomes invalid, you need to bind a new certificate to the port to keep the Scout Keep Alive service running. To do so, on the system the Statistics Service is running on, use the netsh.exe tool of the Windows command-line interface.

> **Note:**

> If the computer has more than one network adapter, the certificate must be bound to all IP addresses.

Viewing the current SSL certificate bindings

1. Launch the command-line interface.
2. Use the following command: `netsh.exe http show sslcert`

All ports with certificate bindings are shown including the relevant information.

Deleting an SSL certificate from a port

1. Launch the command-line interface.
2. Use the netsh.exe tool as shown in the following example: `netsh.exe http delete sslcert ipport=<IP address of host>:22124`

Viewing the thumbprints of certificates

> **Note:**

> Thumbprint corresponds to the certificate hash value.

1. Launch the Powershell. Note that the command is not supported by the normal command-line interface (cmd).
2. Use the following command depending on the certificate store: `dir cert:LocalMachineMy`

For all certificates available in the Microsoft Management Console, the thumbprints are shown under Local ComputerPersonal (with and without binding).

Binding a new SSL certificate to a port

1. Launch the command-line interface.
2. Use the netsh.exe tool with the following command: `netsh.exe http add sslcert ipport=0.0.0.0:22124 certhash=<*thumbprint of your certificate*> appid={957ba029-e2a1-4a13-b426-645a5e3802e2}`

The ipport parameter specifies the IP address and port.

The certhash parameter specifies the thumbprint of the certificate.

The appid parameter is the ID of the Scout Keep Alive service and must not be changed.

Management protocol

Communication between the Scout Server and the eLux® devices can be established via port `22123` or port `22125`.

If you are using a firewall, enable the relevant port.

Certificate-based management protocol

Configuring the trust level on the devices

Configuring Scout Server for communication via CA certificates

Certificate-based management protocol

The certificate-based management protocol provides secure communication between the Scout Server and devices via end-to-end encryption with TLS 1.2.

Starting with Scout 15 2107, only devices with **eLux RP 6.2 or later** are supported. For further information, see Compatibility client platform and Scout Enterprise Management Suite in the Whitepaper **Releases, Lifecycles and Compatibility**.

The certificate-based encryption of the management protocol is carried out via a self-signed certificate automatically generated by the Scout service. Alternatively, you can use a CA certificate that must be configured on the Scout Server.

For the encrypted communication with the Scout Server, port 22125 is used. (Up to version Scout 15 2107, devices with earlier eLux® versions could be used via port 22123 with AES-256 encryption.)

For TLS 1.2 communication, the following requirements must be met:

- On the devices, the trust level must be specified by using **TlsVerifyOption**. By default, the trust level is set to 0 and the certificate check is disabled.

  For further information, see Configuring the trust level on the devices.
- If you are using a certificate issued by a CA (instead of self-signed), the certificate must be provided in the form of a pfx or pem file on the Scout Server. Note that the certificate must not be password-protected. The devices must be equipped with the corresponding root certificates.

  For further information, see Configuring Scout Server for communication via CA certificates.

> **Note:**

> To check the communication via TLS, view the `eluxd.log` log file of the Scout Server service.

Configuring the trust level on the devices

The certificate-based encryption of the management protocol for communication between the Scout Server and the eLux® client requires the verification of the relevant certificates (Chain of trust). By default, encryption is carried out via a self-signed certificate automatically generated by the Scout Server.

> **Important:**

> If you use a certificate issued by a CA (CA certificate), make sure you transfer the corresponding root certificates to the devices. If the root certificate does not exist on a device and certificate check is enabled, the device can no longer be reached by the Scout Server. To

make it easier, you can perform both steps, enabling the certificate check and transferring certificates in one move. **Important:**

In addition, the Scout Server must be configured accordingly and provide the certificate locally.

1. To enable the certificate check, configure the trust level for the relevant devices with the option **TlsVerifyOption**.

   To do so, use the Advanced file entries feature of the Scout Console:

FILE	/SETUP/TERMINAL.INI
Section	Security
Entry	TlsVerifyOption
Value	
0	Certificate is not verified
1	Certificate is verified
3	Certificate is verified with additional verification that the Scout Server name matches the Subject Common Name (CN) or Subject Alternative Name (SAN) in the certificate.

   For further information, see Advanced file entriesin the **Scout** guide.

2. If you use a CA certificate, make sure you transfer all corresponding root/intermediate certificates of your CA to the devices to /setup/cacerts/scoutsrv. This is where the system searches for the required certificates once the certificate check is enabled (Chain of trust).

   For further information, see Files configured for transfer in the **Scout** guide.

3. If you use a CA certificate, in the next step, configure the Scout Server. For further information, see Configuring Scout Server for communication via CA certificates.

4. Restart the devices.

> **Note:**

> After the `terminal.ini` file has been updated on the device, one more device restart might be required to enable the new setting.

> *Once you have enabled trust level 1 or 3 for a device, it can only communicate with its Scout Server by using valid certificates. With trust level 3, the device name is verified in addition.*

Configuring Scout Server for communication via CA certificates

> **Note:**

> This configuration is only required if you are using a certificate that has been issued by a CA.

1. Save the certificate file locally on the Scout Server.
2. On the server machine, in the file system under `%PUBLIC%\Documents\Unicon\Scout\Server` open the `eluxd.ini` file for editing.

   Add the following entries:

SECTION	ENTRY	DESCRIPTION
ELUXD	`UseSelfsignedCertificate=0`	Requires a certificate issued by a CA Use a certificate that is not protected by an additional password. If you set this option (with value 0), you are required to define the next values. Default: 1
ELUXD	`CertificateFile`*=Path to certificate file*	Path to the location of the certificate file Example: `C:\Users\Public\Documents\Unicon\Scout\Server\sampletec-01.pfx`

SECTION	ENTRY	DESCRIPTION
ELUXD	`CertificateKeyFile`= *Path to private key file*	Only required, if the certificate file is not in pfx format

3. Restart the Scout service.
4. Ensure that the certificate check is enabled and the required root/intermediate certificates of your CA are available on the devices (Chain of trust). For further information, see Configuring the trust level on the devices.

From now on, the Scout Server will only communicate with devices that trust the CA certificate.

Troubleshooting

This table lists problems that can occur after you install or update Scout, or when devices download their software, and how to solve them:

ERROR MESSAGE / PROBLEM	REASON	SOLUTION
Scout desktop icons are not displayed correctly.	Desktop icons are cached by Windows. This may lead to the fact that after you update Scout, new icons are not displayed correctly.	Delete the Windows icon cache. Please refer to the relevant documentation.
File access error while checking HTTP/FTP server (error number = 404)	Possibly caused by missing MIME type entries for the used file extensions on the web server	In the MIME type settings of the web server, add the file extensions of the container folders and assign them to the MIME types in the following table. For Microsoft Internet Information Services (IIS), see Example of publishing the containers with IIS in the **UniconOS Management Board** guide.
Devices install an image that Software has taken out of the container folder.	The web server delivers the image file from its cache.	Turn off caching for image files (`.idf`) and signature files (`.sig`) on the web

ERROR MESSAGE / PROBLEM	REASON	SOLUTION
		server. For IIS, see Example of publishing the containers with IIS in the **UniconOS Management Board** guide.

For firmware updates, the web server must deliver the packages, images and signature files of the container folders with these MIME types:

EXTENSION	MIME TYPE
.epm	application/octet-stream
.fpm	application/octet-stream
.idf	application/octet-stream
.sig	application/octet-stream

Don't assign a MIME type to all extensions (`.*`) instead, because the web server would then deliver every file of the container folders.

The following additional assignments might become necessary:

EXTENSION	MIME TYPE	DESCRIPTION
.bin	application/octet-stream	BIOS update via Scout
.bup	text/plain	BIOS update via Scout
.cab	application/vnd.ms-cab-compressed	UEFI update via Scout
.gz	application/x-gzip	Migration from eLux® RP 6 to UniconOS 7: the file `migrationfiles-<version>.tar.gz` in the container folder

EXTENSION	MIME TYPE	DESCRIPTION
.mee	text/plain	Migration of eLux RP 5 devices to eLux RP 6 with whitelist. Different images depending on the hardware model (/en-us/unicon-elux-scout/2611/admin_guides/scout/setup/firmware/firmware_macro_hardware.html)
.rdf	text/plain	Migration from eLux RP 6 to UniconOS 7: the file `recovery_systemEFI.rdf` in the container folder
.udf	text/plain	UEFI updates in analogy to firmware updates (from Scout 15 2107 and eLux RP 6 2107)

Troubleshooting for an installation with Microsoft SQL Server

Scout keeps its license data in the Scout database. When Scout Server starts with a restored or copied database, it binds the license data to this database by itself. This also works if the database is on another SQL Server. You don't need to unlock the database. For this step, Scout Server must be able to read the ID of the database:

ERROR MESSAGE	REASON	SOLUTION
Scout Server stops right after it starts. `eluxd.log` contains `Database GUID is different and the database's own GUID is blank`.	Scout Server can't read the ID of the Scout database. Without this ID, it can't bind the license data to the database.	Contact Citrix® Support.

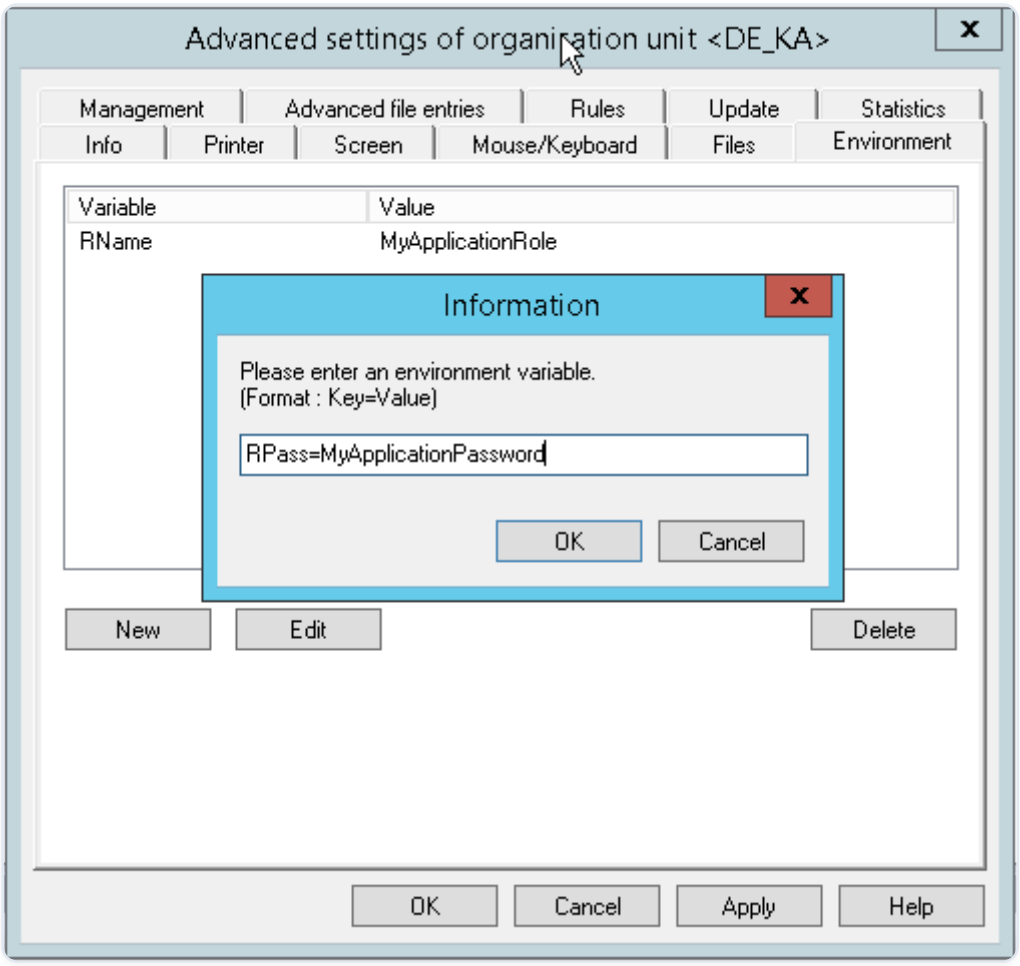
If Scout Server can't connect to the Scout database when it starts, for example because SQL Server isn't running yet after a restart of the computer, Scout Server waits for the database for several minutes. If the database doesn't answer in this time, Scout Server stops:

ERROR MESSAGE	REASON	SOLUTION
Scout Server stops during its start. The Windows Application log shows `Scout Server stopped during start-up: no database connection could be opened (exit code 1017)`.	Scout Server can't connect to the Scout database.	Make sure that SQL Server runs and that Scout Server can reach it. Then start the Scout Server service.

Encrypting values

Whenever you need to encrypt values, we recommend that you add and encrypt variables in a temporary OU, and then copy the encrypted values to their target position.

1. In the Scout Console, create a temporary OU such as `TEMP`.
2. For the TEMP OU, from the context menu, choose **Advanced device configuration > Environment**.
3. Add a new variable and its value. Confirm with OK.



*The new variable and its value are shown in the **Environment** tab.*

4. On the **Environment** tab, right-click the variable, and from the context menu, choose **Encrypt value**.

   The value of the variable is shown in encrypted mode.

5. Select the variable and click **Edit**. Then, copy the encrypted value to the Clipboard and paste it on the target position.
6. Delete the temporary OU.

Appendix

Program and file directories

eLux RP 6 partitions

IP ports

SNMP

Program and file directories

Program directory

The Scout Enterprise Management Suite® by default is installed to

`%PROGRAMFILES%\Unicon\Scout`

Setup installs the Software Management backend to the subfolder `SoftwareManagement` of this directory. The data folder of the backend holds the containers and the log files of the backend. By default, this is `%PROGRAMDATA%\Unicon\Software`.

File path for Scout Server files

Scout log files, configuration files and more are saved to a subdirectory of

`%PUBLIC%\Documents\Unicon`

From UniconOS Management 2611, only administrators can read and change the configuration files in these folders, for example `eluxd.ini` and `install.ini`, so open them in an editor that runs as administrator.

File path for user files

User files are saved to a subdirectory of the local user directory in

`%USERPROFILE%\Documents\Unicon`

Diagnostic files that are requested via UniconOS Management Board are downloaded by the browser used and saved in the download directory, if configured.

> **Note:**

> If you use anti-virus software on your Scout Server, we recommend that you exclude the specified directories from the virus scan to avoid side effects.

eLux® RP 6 Partitions

An eLux device's flash memory is generally divided into three or four partitions when eLux is installed. Each partition is reserved for a dedicated purpose and is only touched when you perform special tasks related to this partition.

All partitions are created during a recovery installation.

PARTITION	REQUIRES	PURPOSE	RECREATED WITH	OTHER
System		Reserved for the firmware (software packages)	Scout Update command with option Format system partition before update	Size for eLux RP 6 2104 LTSR and earlier versions: 1.77 GB / 1.84 GB with/without encryption, Size for eLux RP 6 2107 and later versions: 2.35 GB / 2.41 GB with/without encryption
Boot	only UEFI and USB	Boot section	–	
Setup		Device configuration Local application definitions	Factory reset command	Does not affect the system partition with installed firmware
Update	4 GB flash memory	Software delivery in advance (before firmware update) via Scout command or notification Signature check for eLux software packages	Scout Delivery command with option Format update partition before delivery	The size of the update partition complies with the storage space provided. The update partition is no larger than the storage space provided. Devices with less than 4 GB flash memory are not provided with an Update partition.

Note:

In the Scout Console, in the Properties window of a device, the system, setup, and update partitions are listed including their sizes.

Extended System Partition Starting with eLux RP 6 2107

When you perform an update installation or a new installation (recovery) to eLux RP 6 2107 or later, the system partition is created with 2.35 GB / 2.41 GB (with/without encryption) instead of the previous almost 2.0 GB. This creates more space for the firmware and allows larger images to be used.

UPDATE INSTALLATION

An update installation (firmware update) is still based on the previous partition sizes. The image size is thus still limited to the earlier values. Afterwards, the extended system partition is available and you can install images that may be up to 2.35 GB / 2.41 GB in size. This means, to install larger images on the freshly resized partition of the devices, a second firmware update is required.

RECOVERY INSTALLATION

Provided an up-to-date recovery system is available, with a PXE or USB recovery installation the system partition can be partitioned to the new size directly during the installation process and a larger image with up to 2.35 GB / 2.41 GB can be written in the same process. A new installation or recovery installation thus allows the partition to be resized and used in one step.

DOWNGRADE

To downgrade devices with the extended system partition (eLux RP 6 2107 or later) to an earlier version that only supports the previous system partition with less than 2 GB, you will have to go back to eLux RP 6 2104 LTSR.

We therefore recommend that you update test devices to eLux RP 6 2107 or later as the first step to thoroughly test functionality.

For further information, see Update to new partition layout (1.77 / 1.84 GB to 2.35 / 2.41 GB) in the Scout guide.

Network requirements

All contents relating to ports, protocols and network requirements are covered by the **network requirements** guide. For further information, see Network requirements guide.

SNMP

SNMP (Simple Network Management Protocol) is a network protocol for monitoring and controlling network devices.

For eLux® RP 6 and newer, version SNMPv3 is used.

> **Note:**
>
> The SNMP environment package includes the full net-snmp client toolset, including **snmpget**, **snmpwalk** and **snmpset**, so you can query SNMP status information locally on the device without third party software.

> Configuring SNMP

SNMPD and SNMP Configuration Directives

Configuring SNMP

1. From our myelux.com portal, under **eLux Software Packages**, for your eLux version, under **Add-On**, download the package SNMP Environment and deploy it to the devices.
2. If there is no /setup/snmp/snmpd.conf on the devices, transfer the configuration file snmpd.conf to the devices to `/setup/snmp/snmpd.conf` by using the Scout feature advanced_files Files configured for transfer

   Or:

   Modify the `terminal.ini` file by using the Adding individual file entries feature of Scout. Example:

File	`/setup/terminal.ini`
Section	SNMPD
Entry	rocommunity
Value	secret

3. Optionally, to define further SNMPD and SNMP Configuration Directives, use the Adding individual file entries feature and modify the `terminal.ini` file under SNMPD. Examples:

`syscontact=contact@sampletec.com` `syslocation=testcenter` `doDebugging=1`

For further information on SNMPD Configuration Directives, see http://www.net-snmp.org.

The section `SNMPD` of the `terminal.ini` file is evaluated by the client, and the file `/setup/snmp/snmpd.local.conf` is created, overwriting any existing `/setup/snmp/snmpd.conf`.

If the configuration file does not exist, the file `/setup/snmp/snmpd.local.conf` is created with default values.

NOTES ON CONFIGURING SNMP V3

- When you define users (**createUser**), set a password with at least 8 characters.
- For the authentication method, define `authPriv` or `authNoPriv`.

> **Note:**
>
> For SNMP v2, you can use `noAuthNoPriv` as the authentication method.

SNMPD and SNMP Configuration Directives

The following table refers to the eLux® software package snmp.

For further information on using SNMP with eLux, see SNMP

For further information on SNMP commands, see http://www.net-snmp.org.

APPLICATION	COMMAND
authtrapenable	1 | 2 (1 = enable, 2 = disable)
trapsink	host [community] [port]
trap2sink	host [community] [port]
informsink	host [community] [port]
trapsess	[snmpcmdargs] host
trapcommunity	community-string
agentuser	agentuser

APPLICATION	COMMAND
agentgroup	groupid
agentaddress	SNMP bind address
syslocation	location
syscontact	contact-name
sysservices	NUMBER
interface	name type speed
com2sec	name source community
group	name v1|v2c|usm security
access	name context model level prefix read write notify
view	name type subtree [mask]
rwcommunity	community [default|hostname|network/bits] [oid]
rocommunity	community [default|hostname|network/bits] [oid]
rwuser	user [noauth|auth|priv] [oid]
rouser	user [noauth|auth|priv] [oid]
swap	min-avail
proc	process-name [max-num] [min-num]
procfix	process-name program [arguments...]
pass	miboid command
pass_persist	miboid program
disk	path [minspace | minpercent%]

APPLICATION	COMMAND
load	max1 [max5] [max15]
exec	[miboid] name program arguments
sh	[miboid] name program-or-script arguments
execfix	exec-or-sh-name program [arguments...]
file	file [maxsize]
dlmod	module-name module-path
proxy	[snmpcmd args] host oid [remoteoid]
createUser	username (MD5|SHA) passphrase [DES] [passphrase]
master	specify 'agentx' for AgentX support
engineID	string
engineIDType	num
engineIDNic	string

SNMP CONFIGURATION DIRECTIVES

APPLICATION	COMMAND
doDebugging	(1|0)
debugTokens	token[,token...]
logTimestamp	(1|yes|true|0|no|false)
mibdirs	[mib-dirs|+mib-dirs]
mibs	[mib-tokens|+mib-tokens]
mibfile	mibfile-to-read
showMibErrors	(1|yes|true|0|no|false)

APPLICATION	COMMAND
strictCommentTerm	(1|yes|true|0|no|false)
mibAllowUnderline	(1|yes|true|0|no|false)
mibWarningLevel	integerValue
mibReplaceWithLatest	(1|yes|true|0|no|false)
printNumericEnums	(1|yes|true|0|no|false)
printNumericOids	(1|yes|true|0|no|false)
escapeQuotes	(1|yes|true|0|no|false)
dontBreakdownOids	(1|yes|true|0|no|false)
quickPrinting	(1|yes|true|0|no|false)
numericTimeticks	(1|yes|true|0|no|false)
suffixPrinting	integerValue
extendedIndex	(1|yes|true|0|no|false)
printHexText	(1|yes|true|0|no|false)
dumpPacket	(1|yes|true|0|no|false)
reverseEncodeBER	(1|yes|true|0|no|false)
defaultPort	integerValue
defCommunity	string
noTokenWarnings	(1|yes|true|0|no|false)
noRangeCheck	(1|yes|true|0|no|false)
defSecurityName	string
defContext	string

APPLICATION	COMMAND
defPassphrase	string
defAuthPassphrase	string
defPrivPassphrase	string
defVersion	(1|2c|3)
defAuthType	(MD5|SHA)
defPrivType	(DES)
defSecurityLevel	(noAuthNoPriv|authNoPriv|authPriv)

Network requirements

For deploying eLux® and management solutions, please make sure that the network requirements are implemented. Different products need different protocols and ports for the communication.

eLux

PORT	TYPE	DESCRIPTION	HOW TO DEACTIVATE	COMPONENTS	IN/OUT
	ICMP	ping must be supported to verify the status of the eLux devices		eLux --> Scout Server	Incoming/ Outgoing
80	TCP	Firmware update by using HTTP (and proxy port, if used)		eLux --> Web/File Server, serving the Images and packages.	Outgoing

PORT	TYPE	DESCRIPTION	HOW TO DEACTIVATE	COMPONENTS	IN/OUT
				Best practice: Use HTTPS for a secure connection	
443	TCP	Firmware update via HTTPS/TLS		eLux --> Web/File Server, serving the Images and packages	Outgoing
443	TCP	Onboarding via uPilot		eLux --> `https://onboarding.unicon.com`	Outgoing
5902	WebRTC (TCP, UDP)	Mirroring eLux desktop	In the **Mirroring** section of the device configuration, disable mirroring, or uninstall the VNC server	UniconOS Management Board UI (Web Browser) --> eLux (webrtcify --> VNC Server).	Incoming
				End to end encrypted (WebRTC). High ports (UDP) are used by WebRTC for the data channels.	

PORT	TYPE	DESCRIPTION	HOW TO DEACTIVATE	COMPONENTS	IN/OUT
				You can use the enterprise policy `WebRtcUdpPortRange` (Chrome v54+, Edge v77+) to restrict the local UDP port window (e.g. 10000-11999)	
22124	TCP	Scout Keep Alive Service (optional)		eLux --> Scout Server	Outgoing
22125	TCP	Scout Server (Scout Manager / TLS)		eLux --> SCG, eLux --> Scout Server, Scout Server/SCG --> eLux.	Incoming/ Outgoing
				A certificate can be used with the TlsVerifyOption	
22129	TCP	VPN (SCG)	In SCG Admin Configuration >	eLux --> SCG	Outgoing

PORT	TYPE	DESCRIPTION	HOW TO DEACTIVATE	COMPONENTS	IN/OUT
			Gateways > Public port, you can change the port.		

Note:

Depending on the application and the use-cases, other optional ports might be needed; like DNS, DHCP, TFTP, AD/Kerberos, SNMP, Wake on LAN, etc.

Scout Server

PORT	TYPE	DESCRIPTION	COMPONENTS	IN/OUT
	ICMP	ping must be supported to verify the status of the eLux devices	Scout Server --> eLux	Incoming/ Outgoing
1433	TCP	MS SQL Server	Scout Server --> SQL Server	Outgoing
1434	UDP	MS SQL Server (Browser service)	Scout Server --> SQL Server, Scout Board DBLayer --> SQL Server	Incoming/ Outgoing
22124	TCP (HTTPS)	Scout Keep Alive Service offers an evaluation of configurable status messages of	eLux --> Scout Keep Alive service	Incoming

PORT	TYPE	DESCRIPTION	COMPONENTS	IN/OUT
		the managed devices.		
		These status messages are called keep alive messages.		
		This service runs on the Scout Server		
22125	TCP	eLux Endpoints (Scout Manager / TLS)	Scout Server --> eLux, eLux --> Scout Server	Incoming/ Outgoing
8083 (WSLPORT)	HTTPS	The Citrix License Server port - can be defined/ changed during installation	Scout Server --> Citrix License Server	Outgoing

> **Note:** For details on the ports and firewall requirements, please see Citrix License Server

UniconOS Management Board

PORT	TYPE	DESCRIPTION	COMPONENTS	IN/OUT
1433	TCP	MS SQL Server	Scout Board Database Layer --> SQL Server	Outgoing
5902	WebRTC (TCP, UDP)	Mirroring eLux desktop	UniconOS Management Board --> eLux (webrtcify --> VNC Server)	Outgoing

PORT	TYPE	DESCRIPTION	COMPONENTS	IN/OUT
22160	TCP (HTTPS)	UniconOS Management Board HTTPS web application. Administrator can define a custom port during installation	Administrator --> UniconOS Management Board web UI	Incoming
22150	TCP	Requests of UniconOS Management Board to the Scout Board Database Layer. Needed between computers only if UniconOS Management Board or the Software Management backend runs on another computer than the Database Layer	UniconOS Management Board --> Scout Board Database Layer	Incoming
22151	TCP	Events of the Scout Board Database Layer. With several Scout Servers on one Scout database, the UniconOS Management	UniconOS Management Board --> Scout Board Database Layer	Incoming

PORT	TYPE	DESCRIPTION	COMPONENTS	IN/OUT
		Board of every server connects to the Database Layer of every server		

Software Management backend

The Software Management backend needs these ports only if it runs on another computer than UniconOS Management Board:

PORT	TYPE	DESCRIPTION	COMPONENTS	IN/OUT
22170	TCP (HTTPS)	Software Management backend. The port can be changed during installation	UniconOS Management Board --> Software Management backend	Incoming
22150	TCP	Scout Board Database Layer on the Scout Server	Software Management backend --> Scout Board Database Layer	Outgoing

Scout Cloud Gateway (SCG)

PORT	TYPE	DESCRIPTION	COMPONENTS	IN/OUT
22125	TCP (TLS)	Scout Server (Scout Manager / TLS) - during onboarding only	eLux --> SCG, SCG --> eLux	Incoming/ Outgoing (if 2 network interfaces exist: External Interface)
22129	TCP	VPN (SCG) - can be changed	eLux --> SCG	Incoming (if 2 network interfaces exist:

PORT	TYPE	DESCRIPTION	COMPONENTS	IN/OUT
		in the SCG admin UI		External Interface)
636	TCP	AD Authentication (LDAP)	SCG --> LDAPS (AD)	Outgoing (if 2 network interfaces exist: Internal Interface)

Licensing Guide

The Licensing Guide explains how UniconOS Management (Scout) is licensed through a Citrix® License Server and how you get your entitlements from Citrix.

See the licensing guide for detailed information.

Theming guide for eLux 7

The look and feel of eLux 7 can be modified to realize corporate identity requirements to some extent.

This guide shows,

- How to create themes for eLux 7,
- Where to place the required files and configurations,
- How to activate a theme

Components of a theme

A theme consists of a couple of predefined components. The configuration and the naming of the files follow predefined conventions.

Creating a custom theme

To create a custom theme, follow the following steps:

Create the theme configuration file

Each theme requires a configuration file for the theme, called `config.json`.

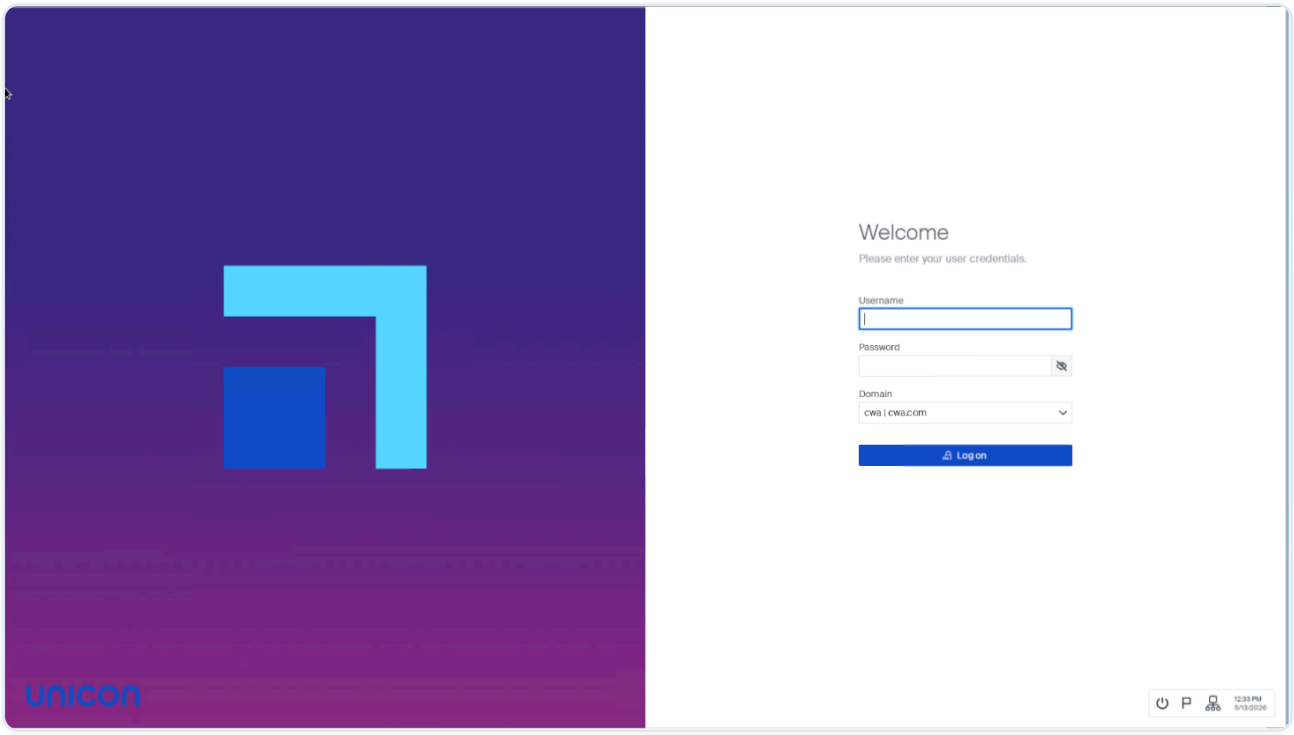
```
{
  "name": "my-theme",
  "accentColor": "#CD5767",
  "desktopBackgroundImage": {"type": "jpg", "scaling": "crop"},
  "productImage": {"type": "png", "scaling": "contain"},
  "brandImage": {"type": "png", "scaling": "contain"},
  "welcomeBackgroundImage": {"type": "png", "scaling": "crop"},
  "welcomeBackgroundLayout": "leftImage"
}
```

Note:

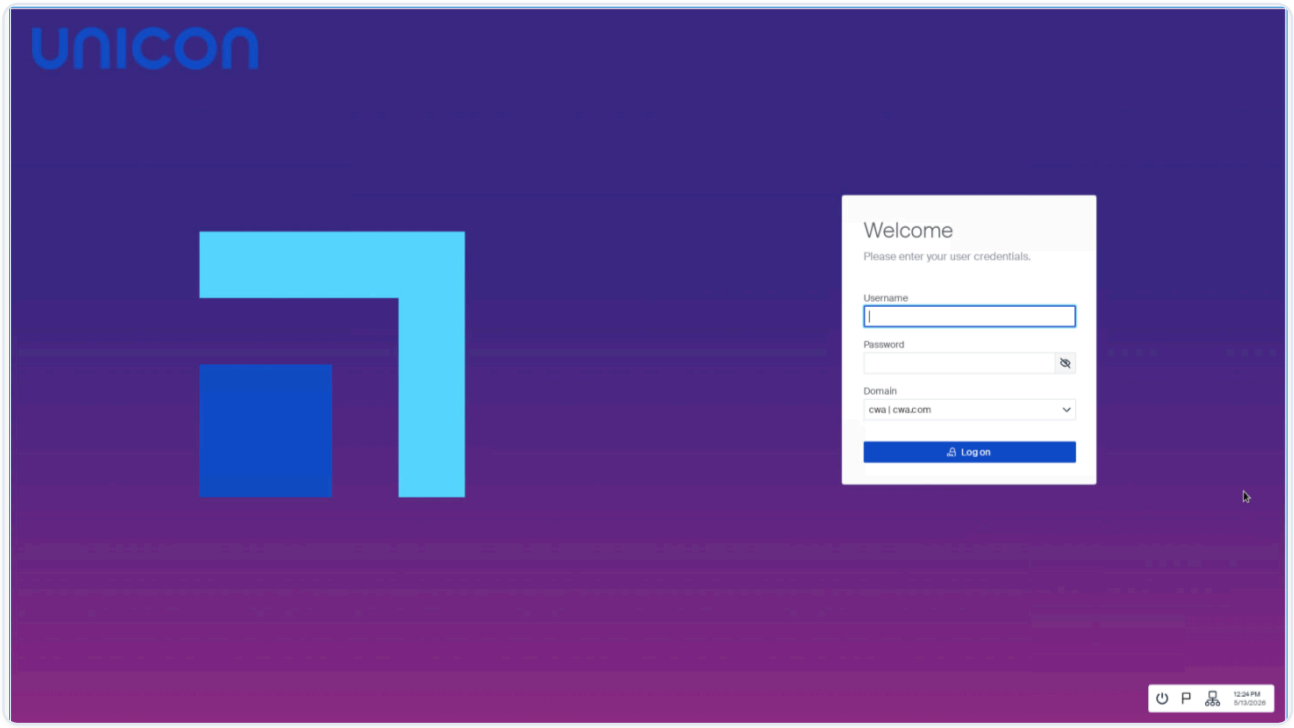
The name of the theme and the accent color need to be defined by the theme requirements. The config file must be named `config.json` by convention.

The `welcomeBackgroundLayout` field controls how the welcome background image is displayed on the login screen, lock screen, and welcome screen. The following modes are available:

- **`"none"`** (default) — Standard layout. No background image on the login screen.
- **`"leftImage"`** — The welcome background image covers the left half of the screen. The login form remains on the right side.



- `"fullImage"` — The welcome background image covers the entire screen. The login form appears in a floating card on the right half.



Prepare the theme files

Prepare the image files for the desktop background, product image, brand image, and optionally the welcome background image, according to the requirements.

> **Note:**

The file names for the images must be named `desktopBackgroundImage.jpg`, `productImage.png`, `brandImage.png`, and `welcomeBackgroundImage.png` by convention.

Transfer the theme files to eLux 7

Transfer all the theme related files to eLux 7 in UniconOS Management Board, using the advanced file entries. The theme files will all go into this theme folder.

SOURCE FILE	DESTINATION
`config.json`	`/setup/elux/themes/my-theme/config.json`
`desktopBackgroundImage.jpg`	`/setup/elux/themes/my-theme/desktopBackgroundImage.jpg`
`productImage.png`	`/setup/elux/themes/my-theme/productImage.png`
`brandImage.png`	`/setup/elux/themes/my-theme/brandImage.png`
`welcomeBackgroundImage.png`	`/setup/elux/themes/my-theme/welcomeBackgroundImage.png`

Note:

After applying the configuration, reboot the endpoint, so that all the configuration and theme files are being transferred. The `welcomeBackgroundImage` file is only required if `welcomeBackgroundLayout` is set to `"leftImage"` or `"fullImage"`.

The eLux default theme with all its files is located at `/setup/elux/themes/unicon_default` in case a template is needed. Also, when a device factory reset is performed, eLux will revert to the default theme.

Set the theme

Once you have transferred all the required files to eLux 7, the theme needs to be activated.

The theme can be set via the `terminal.ini` configuration file. To do so, use the Scout feature Advanced file entries:

FIELD	VALUE
File	`/setup/terminal.ini`
Section	`Layout`
Entry	`CurrentTheme`
Value	`my-theme`

After applying the configuration, reboot the endpoint for the theme to take effect.

Overview

eLux® Portable is an ideal solution for secure work-from-home and remote access environments. It is compatible with x86 endpoints that meet the minimum hardware requirements for UniconOS. The key benefits for the users are that it is immediately available without any installation efforts. So it is a real Plug & Work approach.

Features

- Any home PC equipment or other x86 device with a standard USB port can be used without modifying the underlying operating system.
- Any state-of-the-art USB drive with a capacity larger than 4 GB can be used.
- No company data are stored on the USB drive.
- No need for special security measures or dedicated hardware
- No installation
- The First Configuration Wizard guides you through the first steps, no additional configuration needed.
- The license always remains on the server, so a broken USB drive is easily replaced and remote work can continue seamlessly.

Requirements

- A stable internet connection
- On your device, change the boot order for boot from USB and disable secure boot.

Creating the eLux® Portable stick

REQUIRES
• PC running Windows 10 or later versions
• USB stick on local port with 4 GB memory space

1. From our myelux.com portal, under **Downloads > eLux > eLux Portable**, download the current **eLux Portable** archive.
2. Unpack the downloaded .zip file on your device.
    - The eLux USB Stick image is available as a .dd file in a directory on your PC, along with the StickWizz.exe program (and a describing text file with check sum).
3. Continue with Writing an image to USB.

Writing an image to USB (with StickWizz)

REQUIRES
• Image file as a .stw or .dd file on a PC running Windows 10 or later versions
• StickWizz application on your PC (/en-us/unicon-elux-scout/2611/short_guides/recovery_general/recovery_usb/recovery_usb_prepare.html)
• USB stick on local port with 4 GB memory space, write access

1. Plug the USB stick into the PC. Note that the Windows Explorer must not be open on the USB drive.
2. Run the StickWizz.exe program.

- The language depends on the Windows display language selected.
- The checksum is calculated and compared to the one provided.

3. Check whether the appropriate source under Image file (1) and the appropriate destination under Device (3) are selected.
4. Click **Write** (7).

The image is written to the USB stick. A message confirms the successful completion.

1	After StickWizz is started, it shows a .stw or .dd file located in the same directory.
2	Browse to select your image file from another directory, if required.
3	Device /drive, on which a USB stick is connected. Choose a different drive if required.
4	Delete all data from the selected device (before writing the image).
5	During the write process, each written data packet will be compared to the source.
6	As soon as StickWizz opens with the specified file, the checksum is calculated and displayed.
7	Start writing.

Using eLux® Portable

REQUIRES
PC, notebook or other x86 device providing a boot option from **USB HDD**

Starting eLux from USB

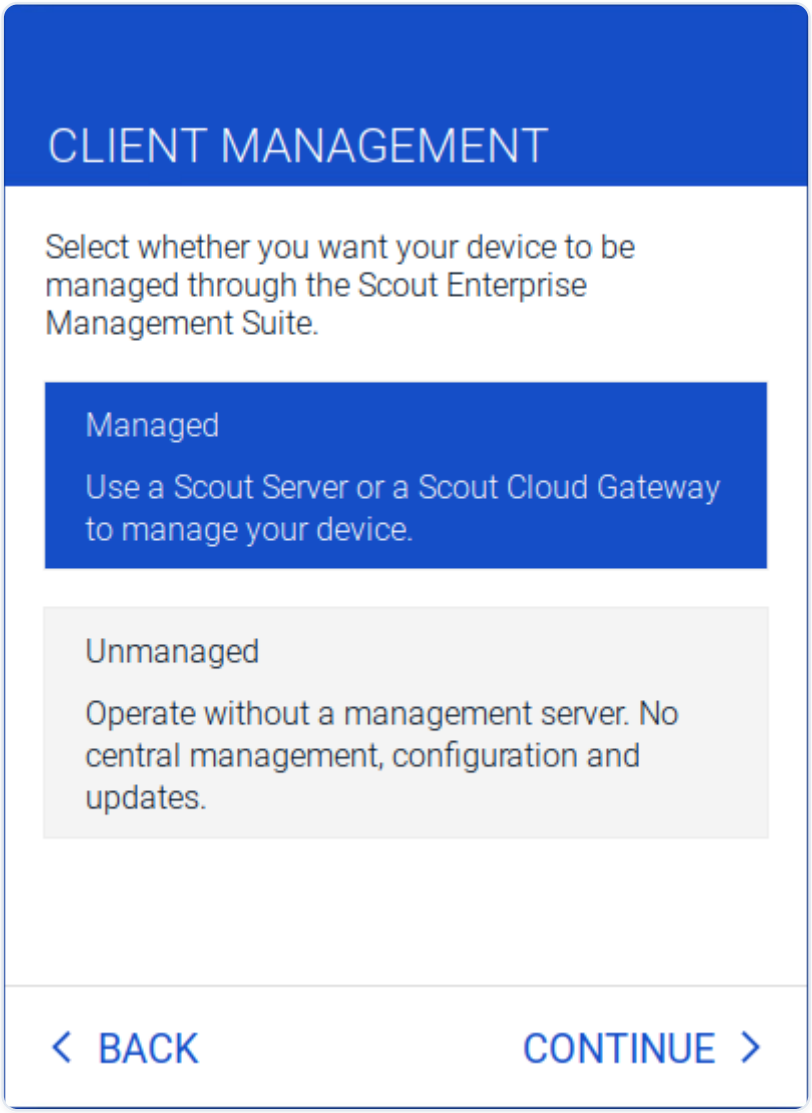
1. Plug the USB stick into a USB port and boot the device.
2. To enter the BIOS/UEFI menu, interrupt the boot process by pressing ESC, F2, F12, or the key specified by the manufacturer.

3. Configure the BIOS/UEFI of your device to boot from USB. Disable **Secure boot**.
4. Restart the device.

The device boots from the USB stick and eLux loads with the First Configuration Wizard.

Configure access to your infrastructure

1. In the First Configuration Wizard, select your preferred display language and keyboard language.
2. Agree to the license agreement.
3. Enter the data of your WLAN or your LAN connection..
4. To configure client management via Scout, select **Managed**.



5. Enter the address of your Scout Server (or Scout Cloud Gateway) as FQDN or IP address.
6. Enter your logon data.
7. Click **Finish**.

Your device connects to its Scout Server and your desktop is loaded. The connection data is retained, so you will access your eLux desktop without reconfiguration after the next device restart from USB.

> **Note:**
>
> The operating system and the data saved on your hard disk will remain completely unaffected.

Overview

A recovery installation allows you to install the eLux operating system on a device's flash memory and/or to reset the device to the initial state.

In the following cases a recovery installation can be suitable:

- The client does not boot anymore (=> restore)
- The flash memory of the client does not provide an image/is empty
- The local device password was lost
- The operating system on the flash memory shall be replaced by eLux®
- A factory reset of the image on the flash memory is required
- Migration to a later eLux version (alternatively see the **Migration** short guide)
- If critical feature-packages `(.fpm)` in the Base OS are updated, the client requires a recovery installation before an update can be performed.

> **Note:**
>
> A recovery installation will overwrite all data on the storage medium and install the eLux software. It cannot be undone.

> A recovery installation can be performed in two different ways:

- via USB stick if the client hardware supports booting from USB mass storage devices
- via network if the client's network adapter supports PXE

Recovery via USB

On our myelux.com portal, you can download **eLux USB Stick** images for all supported eLux versions. By using the tool included in the .zip file, transfer the image file to a USB stick and create an eLux recovery stick which you can use for installing eLux on your devices.

To recover your individual firmware image, export it from UniconOS Management Board as a recovery stick file (`.stw`), and create a USB stick with it. In **Software**, under **Images**, open the menu of your image, and click **Export recovery stick (StickWizz)**. For further information, see Further actions for an image in the **UniconOS Management Board** guide.

The `.stw` file also contains the newest recovery image (`.dd`) of the container, which makes the stick bootable. The export works only for containers of UniconOS 7 that hold a recovery image. From UniconOS 2611 and UniconOS 2603 LTSR CU3, a container that you fill from the AllPackages archive already has this recovery image.

- Preparing a recovery image for USB
- Writing an image to USB (with StickWizz)
- Performing a USB recovery

Preparing a recovery image for USB

REQUIRES
PC running Windows 10 or later versions

eLux® USB Stick Image

1. From our myelux.com portal, under **eLux USB Stick Images**, download the package for the required eLux version.
2. Unpack the downloaded `.zip` file.

The eLux USB Stick image is available as a .dd file in a directory on your PC, along with the StickWizz.exe program (and a describing text file with check sum).

Individual Image

1. In the **Navigation** menu of UniconOS Management Board, click **Software**, and then click **Images** in the **Software** menu.
2. In the image list, open the menu of your image, and click **Export recovery stick (StickWizz)**.

Software downloads a `.stw` file with the image, its packages and the newest recovery image (`.dd`) of the container. This works only for containers of UniconOS 7 that hold a recovery image. For further information, see Further actions for an image in the **UniconOS Management Board** guide.

3. Save the `.stw` file on your PC.
4. From our myelux.com portal, under **Tools**, download the **StickWizz** program , and then unpack the `.zip` file on your PC.

Your individual image is available as a `.stw` file in a directory on your PC, along with the `StickWizz.exe` program. Continue with Writing an image to USB.

Writing an image to USB (with StickWizz)

REQUIRES
Image file as a .stw or .dd file on a PC running Windows 10 or later versions
StickWizz application on your PC (recovery_usb_prepare)
USB stick on local port with 4 GB memory space, write access

1. Plug the USB stick into the PC. Note that the Windows Explorer must not be open on the USB drive.
2. Run the `StickWizz.exe` program.
    - The language depends on the Windows display language selected.
    - The checksum is calculated and compared to the one provided.
3. Check whether the appropriate source under **Image file** (1) and the appropriate destination under **Device** (3) are selected.
4. Click **Write** (7).

The image is written to the USB stick. A message confirms the successful completion.

1	After StickWizz is started, it shows a .stw or .dd file located in the same directory.

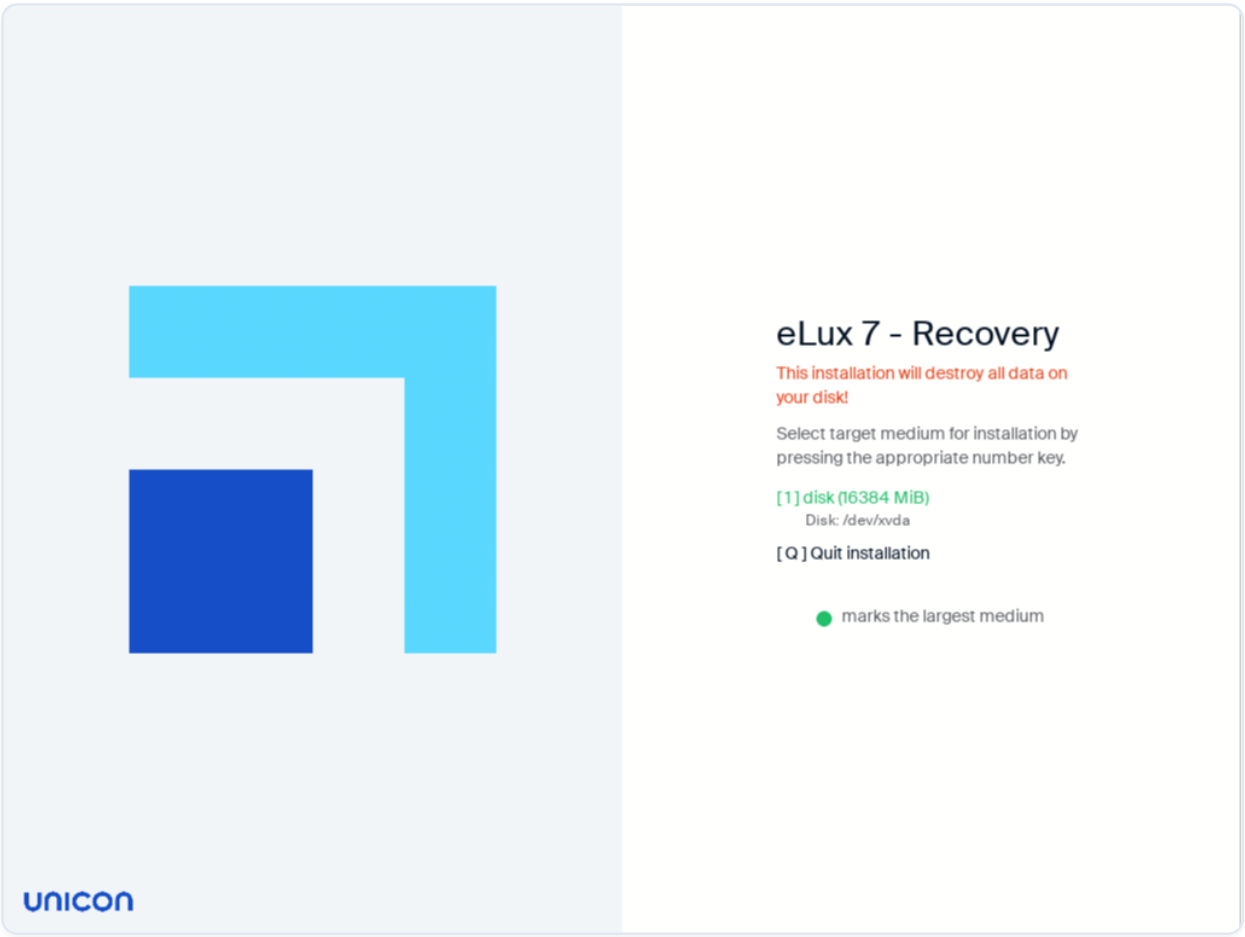
2	Browse to select your image file from another directory, if required.
3	Device /drive, on which a USB stick is connected. Choose a different drive if required.
4	Delete all data from the selected device (before writing the image).
5	During the write process, each written data packet will be compared to the source.
6	As soon as StickWizz opens with the specified file, the checksum is calculated and displayed.
7	Start writing.

Performing a USB recovery

REQUIRES
• *PC, mobile device or other client device providing a boot option from **USB HDD**
• 4 GB (2 GB for limited functionality) memory space on the flash memory / hard disk

1. Configure the BIOS of your device for boot from USB.
2. Connect the USB stick to a USB port and boot the device.
3. To call the boot menu, during the boot process, press F12 or the relevant key specified by the supplier.
4. On the eLux® boot menu, select the Install eLux XX option.
5. Select the target medium for the installation by pressing the number key shown next to it.

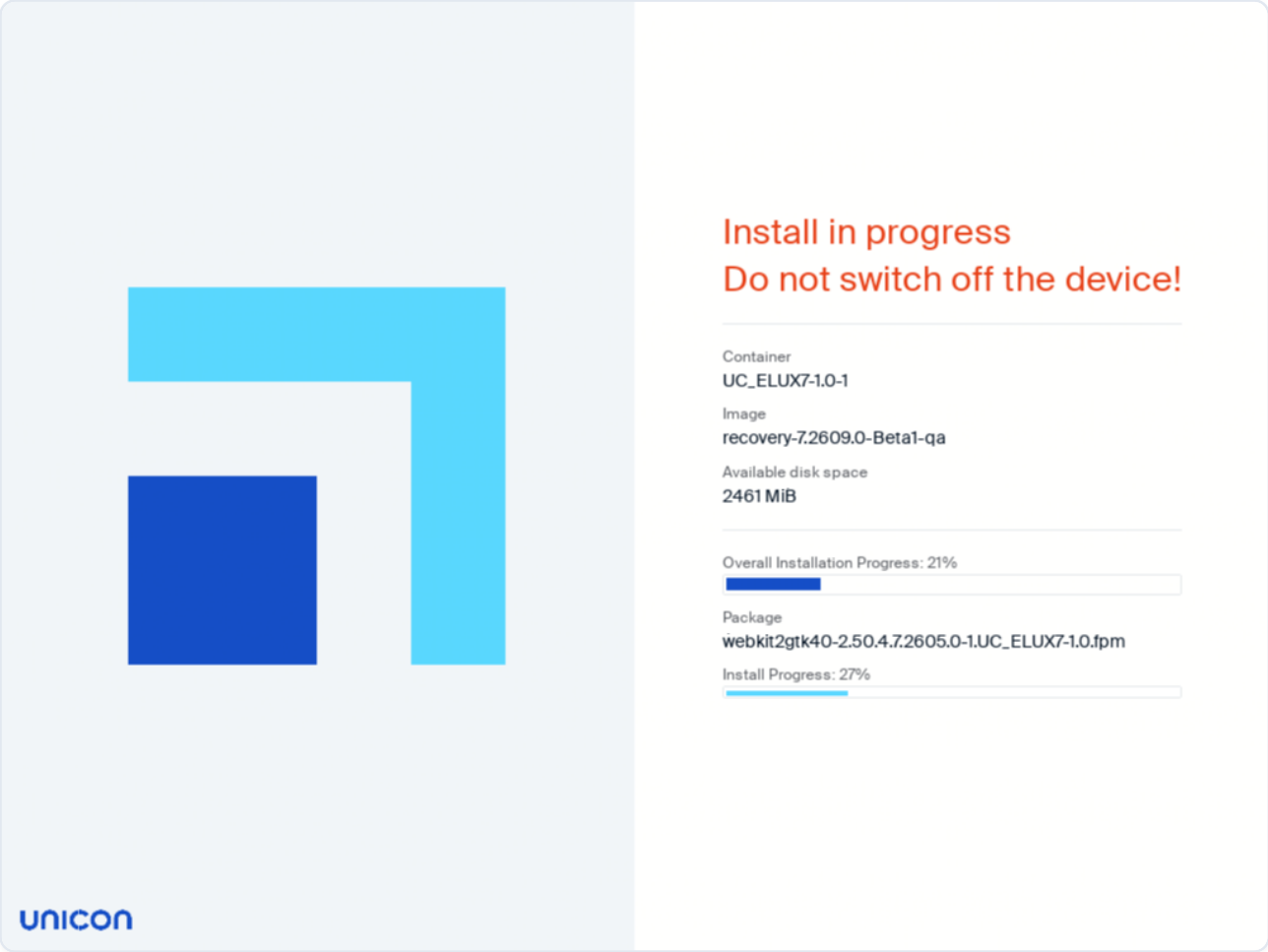
   The medium marked with a green dot is the largest one found on the device.



Important:

During the installation process, all data and the entire operating system are deleted from the hard disk (system recovery). All eLux and Management license information is kept as is.

The installation starts and shows the current package being installed and the overall progress. Do not switch off the device during this process.



After the installation is completed, the client restarts. Depending on the network configuration, the device is automatically assigned to a Scout Server via DHCP or DNS, or the First Configuration Wizard starts.

Network recovery (PXE)

The PXE network protocol (Preboot Execution Environment) enables devices to boot via LAN. The devices do not need a bootable storage medium but their BIOS must support PXE.

The PXE code in the BIOS of the network adapter allows the devices to communicate with the DHCP and TFTP server. From the DHCP server, the client receives the IP configuration. The TFTP server provides the boot file. The boot file calls the core installer which installs a minimal system, formats the flash memory and prepares the installation of the eLux® operating system from the web server.

With the **Recovery service** of the Scout Enterprise Management Suite®, a TFTP server and the required functions are installed. The **Recovery service** is included in the standard installation.

- PXE recovery requirements
- Configuring PXE recovery

- DHCP configuration
- Performing a PXE recovery
- Troubleshooting

PXE recovery requirements

- The relevant devices are provided with a PXE-capable BIOS and a LAN connection. UEFI devices can be installed via a PXE recovery.
- Scout Enterprise Management Suite® including the **Recovery service** (included in standard installation)
- eLux® software container for the relevant eLux version, installed on an HTTP or FTP server
- DHCP server (included in Microsoft Windows Server operating system)

The TFTP server is installed and registered as a service by the Scout Enterprise Management Suite. Make sure that there is no other TFTP server running on the system.

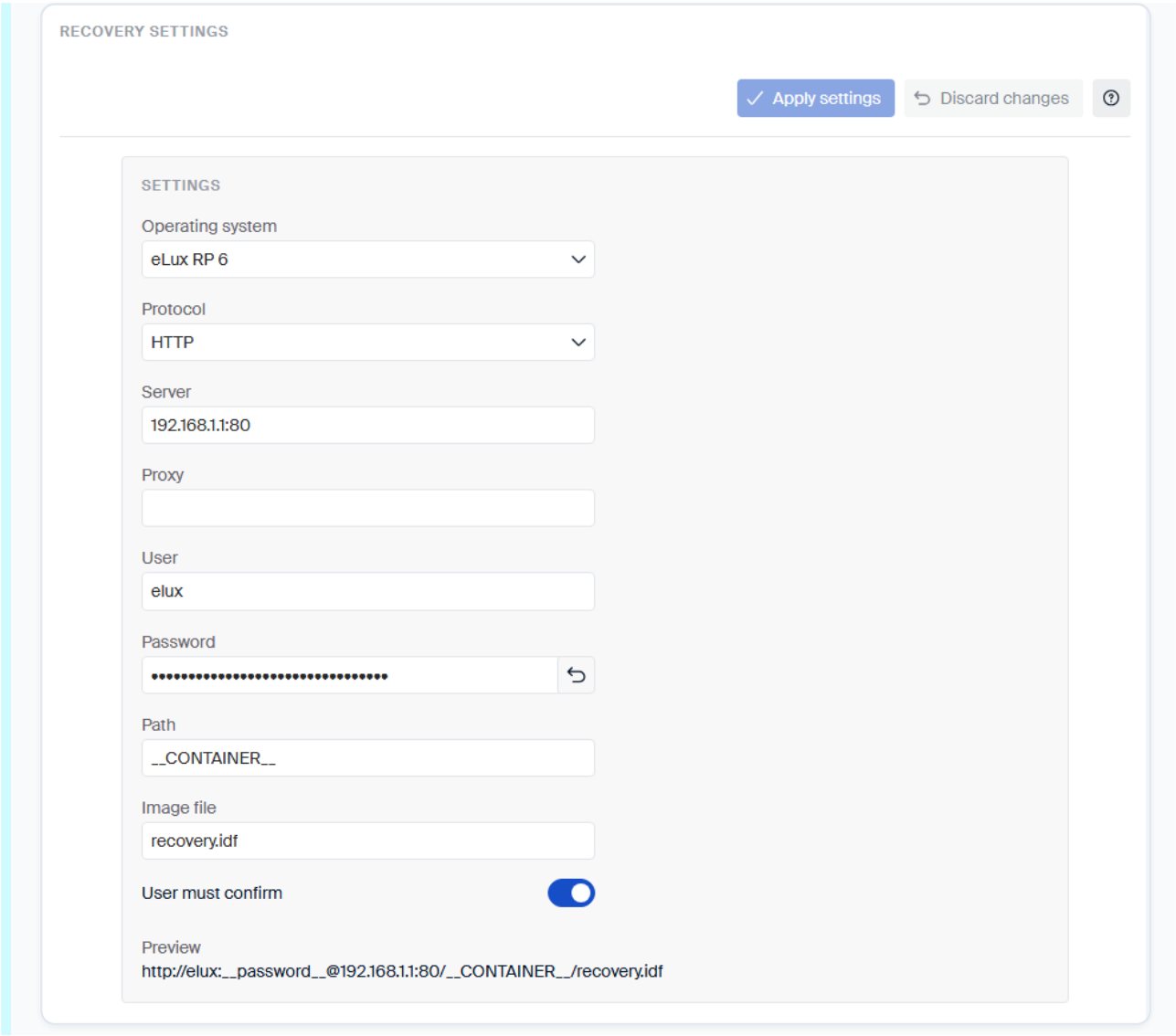
Configuring PXE recovery

Recovery settings is where you tell devices which server to download the eLux® software packages from during a PXE recovery.

1. In UniconOS Management Board, click **Preferences**, and then click **Recovery settings**.
2. In the **Settings** section, fill in the fields that the table below describes.
3. Click **Apply settings**.

> **Note:**

> In the legacy Scout Console, you find the settings under **Options** > **Recovery settings**. Confirm with **OK** instead of **Apply settings**.



These fields describe the server that provides the software packages and how devices connect to it:

FIELD	TYPE	DESCRIPTION
Operating system	Dropdown	The eLux version of the devices you want to recover: **eLux RP 6** or **eLux 7**.
Protocol	Dropdown	How devices download the software packages: **HTTP** or **FTP**.
Server	Text field	The name (FQDN) or IP address of the server, with an optional port. Example: `192.168.10.1:80`

FIELD	TYPE	DESCRIPTION
Proxy	Text field	Optional. The IP address and port of a proxy server. Example: `192.168.10.2:3128`
User	Text field	Optional. The user name for the server, if it requires one.
Password	Text field	Optional. The password that belongs to that user name.
Path	Text field	The folder on the server that holds the software packages. See Path.
Image file	Text field	The image definition file (IDF) to use. See Image file.
User must confirm	Toggle	Turn on to make users confirm on the device before the recovery installation starts.

Preview, below the fields, shows the address that your entries produce. It doesn't show the password.

Path

Separate the folders with `/`. Which path you enter depends on the ELIAS version that provides the packages:

- With ELIAS 18, enter the path name that you defined during installation, for example `elias/UC_ELUX`.
- With the legacy ELIAS, enter `eluxng/<container name>`.

To use one setting for different major eLux versions, enter the `__CONTAINER__` parameter. For details, see Different eLux versions in the **Scout** guide.

Image file

The image definition file defines which software packages devices install. Enter its file name, for example `recovery.idf`. For details, see Firmware tab in the **Scout** guide.

To update devices that have an earlier partition layout to a version with an extended system partition, see New partition layout in the **Scout** guide.

DHCP configuration

Next to selecting DHCP options for the router, DNS server and domain name which are already required for eLux®, for a PXE recovery installation, you need to define the boot server and boot file as DHCP options.

1. Open the DHCP manager as administrator.
2. For the relevant DHCP server, select either the **Server options, the Scope Options or the Reservations** for individual devices, and then click **Action > Configure Options... > General**.
3. Edit the following DHCP options:

OPTION	VALUE
066 Boot Server Host Name	`<IP address of the TFTP server>` (normally same as Scout Server IP address)
067 Bootfile Name	grub/pxex64.efi for UEFI devices (x64) grub/pxe.i386 for non-UEFI devices

4. Confirm with **OK**.

Performing a PXE recovery

The recovery installation is initiated by the client. The client is required to use the boot medium **LAN** and, depending on the hardware, must be configured accordingly.

- Turn the device on and then, on the boot menu, select the boot medium option LAN.

or

- If the boot menu does not provide boot medium options, in the BIOS, configure the **First boot device** to LAN and then restart the device.

After the core installer has prepared the system, the software packages are transferred from the web server and are installed. After the installation is completed, the client restarts. Depending on

the network configuration, the device is automatically assigned to a Scout Server via DHCP or DNS, or the First Configuration Wizard starts.

Troubleshooting

Consulting the server log files helps you analyze problems with the recovery installation:

- `tftpd.log` to be found in `Documents\UniCon\Scout\Recovery`
- DHCP server log file

While the software packages are installed, a progress bar is shown on the screen. To switch from graphic mode to text mode, press CTRL+ALT+F4 to read texts or any error messages.

For problems and error messages, see the following table.

PROBLEM	REASON	SOLUTION
PXE recovery cannot be used for UEFI devices.	The eLux version being used does not support PXE recovery for UEFI devices.	Use current eLux and Scout versions. For further information, see DHCP configuration.
When starting the PXE recovery, the device reports a DHCP timeout.	The DHCP server does not respond.	*Check the network connection. In the DHCP server log file, check whether the client has received an IP address. Modify your DHCP server if necessary.
The device starts the PXE recovery but then boots normally or reports a TFTP timeout: TFTP open timeout	The TFTP server does not respond.	Check if the TFTP service is available. Analyze the log file of the TFTPD daemon. Check the entries for the router/ gateway and boot server of your DHCP service.
After starting the PXE recovery, the device reports a TFTP error: TFTP Error – File not found	The TFTP server failed to send the boot file (pxelinux.0)	Check the boot file entry of your DHCP server. Analyze the TFTP log file. Check the access rights for the TFTP server root directory.

PROBLEM	REASON	SOLUTION
The recovery stops and a black screen is displayed with the following message: could not find image: eluxrp.krnboot	The TFTP server failed to provide the eluxrp.krn file.	Analyze the TFTP log file.* Check the access rights and whether the recovery files are available in the TFTP server root directory.
The recovery stops and a black screen is displayed with the following message: could not find ramdisk image: ramfs.rp boot	The TFTP server failed to provide the ramfs.rp file.	Analyze the TFTP log file.* Check the access rights and if the recovery files are available in the TFTP server root directory.
The recovery stops after the following messages are displayed: `ec = 406 ... elux-library.... failed` `http://user:password@webserverorfailed` `ftp://user:password@ftpserver`	The recovery image file cannot be transferred from the web or FTP server.	Wait for the FTP or HTTP timeout to occur and check whether the displayed address is valid: `http://user:passwort@webserver` or `ftp://user:passwort@ftpserver`. Modify the values in the Recovery settings of UniconOS Management Board if necessary.

eLux® Diagnostic Tools

eLux includes a set of Linux diagnostics tools — `lsof`, `fuser`, `ping`, and `tcpdump` — for investigating system and networking issues directly on the device, without escalating to a support case. It also supports automatically recording network traffic from the moment a device boots, for troubleshooting connection problems that happen before anyone can even log in.

Available Tools

Four command-line tools are available for investigating problems directly on the device:

TOOL	PURPOSE	ACCESS
`ping`	Checks whether another device or server can be reached over the network. A	Non-root `elux` user

TOOL	PURPOSE	ACCESS
	full iputils binary, not the busybox applet.	
`lsof`	Lists open files and file descriptors — useful for seeing which running program has a particular file open.	Non-root `elux` user
`fuser`	Identifies which process is using a file or socket — useful before unmounting or deleting something.	Non-root `elux` user
`tcpdump`	Records raw network traffic (a "packet capture") so you can see exactly what data was sent and received.	Root only, by design

All four are available since **eLux 7.2607**, but two need an extra package before they'll show up on a device:

- `ping` ships in the base `baseos` package — no extra step needed.
- `lsof` and `fuser` need the `baseos_extra_tools` feature package (FPM).
- `tcpdump` needs the `admin_tools` package, feature package `admin_tools_tcpdump_boottime`.
- Adding these feature packages may require changing the image in **Software** in UniconOS Management Board, see Changing an image.

Boot-Time Network Capture

Instead of waiting for someone to notice a problem and manually start a capture, eLux can record network traffic automatically, starting from the moment the device powers on — early enough to catch connectivity or configuration issues that happen before a user can even log in, for example during 802.1X authentication. While a capture is running, a desktop notification tells the signed-in user their traffic is being recorded.

Set the following in `/setup/terminal.ini`, section `Global`, using **Scout → Advanced device configuration → Advanced file entries**:

KEY	VALUE	DESCRIPTION
`BootTimeNetCapture`	`true`	Enables boot-time capture. Case-sensitive — only lowercase `true` is accepted; any other value, including `TRUE`, is treated as disabled, and the device still boots normally.
`BootTimeNetCaptureMaxFileSize`	Size in MB, for example `10`	Maximum size per capture file before it rotates.
`BootTimeNetCaptureMaxFileCount`	Number of files, for example `2`	Maximum number of rotating capture files kept at once.

Add each entry, then click **Apply** and reboot the device.

Capture files are written to `/tmp/boottime.pcap0`, `/tmp/boottime.pcap1`, and so on, rotating once a file reaches the configured size — the oldest file is overwritten once the file count limit is reached. Capture continues seamlessly across LAN/WLAN interface switches.

Useful Commands

WHAT	COMMAND
Check tool availability	`which ping lsof fuser tcpdump`
Confirm ping is the real binary, not busybox	`file $(which ping)`
List boot-time capture files	`ls -la /tmp/boottime.pcap*`
Check whether boot-time capture is running	`ps aux | grep tcpdump`
Verify boot-time capture settings	`ucsettingsctl -o Global`
Live capture (root)	`tcpdump -i any -c 10`
Capture to file (root)	`tcpdump -i any -w /tmp/networktrace.pcap -c 50`

WHAT	COMMAND
Read a capture file	`tcpdump -r /tmp/networktrace.pcap | head`
List open files for a path	`lsof /home/elux`
List network connections	`lsof -i`
Show processes using a path	`fuser /home/elux/`

Good to Know

- Capture files in `/tmp` are owned by **root** — exporting one to USB requires `sudo cp`.
- Boot-time capture files are included in Scout diagnostics packages requested from the device, and can be opened and analyzed in Wireshark.
- Running `tcpdump` as the non-root `elux` user fails with `Operation not permitted` — expected, not a bug (see Access above).
- If `which tcpdump` returns empty, that's expected on a device without the `admin_tools` package — not a fault.

IEEE 802.1X authentication for eLux

Certificate-based logon with 802.1X

IEEE 802.1X is an IEEE standard for port-based Network Access Control in IEEE 802 networks. It provides an authentication mechanism for client devices (supplicants) wishing to attach to a LAN or WLAN. The supplicant provides credentials such as a digital certificate to an authenticator, and the authenticator forwards the credentials to an authentication server (RADIUS) for verification. The authenticator can be an IEEE 802.1X-capable Ethernet switch or wireless access point. If the authentication server determines the credentials are valid, the supplicant is allowed to access the protected side of the network.

As a RADIUS server, you can use the Microsoft Network Policy Server (NPS) or a freeware program such as freeRADIUS.

The supplicant is implemented as a software program. We support the free software wpa_supplicant.

The standard recommends the Extensible Authentication Protocol (EAP) or the PPP-EAP-TLS Authentication Protocol for authentication.

> **Note:**
>
> For certificate enrollment and management, you can use SCEP which is supported by eLux.

Certificate-based logon with 802.1X and TPM 2.0

In combination with SCEP, private keys on TPM 2.0 devices can be stored in the TPM 2.0 module. To further secure authentication, starting with eLux RP 6.2021, you have the option to have cryptographic keys generated inside the TPM 2.0 chip of a device. These keys can neither be displayed nor exported.

For further information, see Certificates for SCEP in the SCEP guide.

IEEE 802.1X authentication for eLux

Certificate-based logon with 802.1X

IEEE 802.1X is an IEEE standard for port-based Network Access Control in IEEE 802 networks. It provides an authentication mechanism for client devices (supplicants) wishing to attach to a LAN or WLAN. The supplicant provides credentials such as a digital certificate to an authenticator, and the authenticator forwards the credentials to an authentication server (RADIUS) for verification. The authenticator can be an IEEE 802.1X-capable Ethernet switch or wireless access point. If the authentication server determines the credentials are valid, the supplicant is allowed to access the protected side of the network.

As a RADIUS server, you can use the Microsoft Network Policy Server (NPS) or a freeware program such as freeRADIUS.

The supplicant is implemented as a software program. We support the free software **wpa_supplicant**.

The standard recommends the Extensible Authentication Protocol (EAP) or the PPP-EAP-TLS Authentication Protocol for authentication.

> **Note:**

> For certificate enrollment and management, you can use SCEP which is supported by eLux.

Certificate-based logon with 802.1X and TPM 2.0

In combination with SCEP, private keys on TPM 2.0 devices can be stored in the TPM 2.0 module. To further secure authentication, starting with eLux RP 6.2021, you have the option to have cryptographic keys generated inside the TPM 2.0 chip of a device. These keys can neither be displayed nor exported.

For further information, see Certificates for SCEP in the SCEP guide.

Configuring 802.1X for eLux®

> **Note:**
>
> 802.1X can be configured for LAN or WLAN.The procedure differs in the location of the configuration file and in some parameters.For further information, see Configuring WPA supplicant.

1. Make sure that the eLux package **WLAN drivers** and the included feature package **WPA supplicant** are installed on the devices. This may require changing the image in **Software** in UniconOS Management Board, see Changing an image.
2. Transfer the required certificates to the devices to /setup/cacerts by using the Scout feature Files configured for transfer.

   > **Note:**
   >
   > The certificates of your RADIUS environment require the FQDN for the Common Name (CN).

3. Configure the file wpa.conf, and then transfer the configuration file to the devices by using the Scout feature Files configured for transfer.For further information, see Configuring WPA supplicant.
4. If you use SCEP, alternatively generate the file wpa.conf from the template wpa.conf.scep.For further information, see WPA configuration via template.

5. Only LAN: In UniconOS Management Board, open the device configuration of the relevant OU. In the **Network** section, under **LAN**, click **Edit LAN connection**. In the **IEEE 802.1X authentication** group, select **Activate authentication via IEEE 802.1X**, see IEEE 802.1X authentication in the **UniconOS Management Board** guide.

If the configuration is correct and the required certificates are rolled out, you can use devices on the 802.1X port.

Choosing Between SCEP and EST for LAN

For **LAN** 802.1X, a client certificate can come from either SCEP or EST. If only one is enabled, that source is used automatically. If both are enabled, `PreferForLanConfig` decides which one wins — a boolean in the `[Certificate]` section of `/setup/scep/scep.ini` and `/setup/est/est.ini`, `false` by default.

SCEP	EST	`PREFERFORLANCONFIG`	SOURCE USED	WPA.CONF LOADED
On	Off	–	SCEP	`/setup/scep/wpa.conf`
Off	On	–	EST	`/setup/est/wpa.conf`
On	On	`false` (default)	SCEP	`/setup/scep/wpa.conf`
On	On	`true`	EST	`/setup/est/wpa.conf`

To use EST instead of SCEP when both are enabled, set `PreferForLanConfig=true` in `est.ini`. Setting it to `true` in both `scep.ini` and `est.ini` is a misconfiguration — SCEP wins and an error is logged.

Enabling or changing SCEP or EST requires a device restart.

Configuring WPA supplicant

> **Note:**

> You can use the example files on the devices to configure the WPA supplicant: `/setup/scep/wpa.conf`.

1. Create an individual wpa.conf configuration file.

   By default, the file contains the following information:

```
ctrl_interface=/var/run/wpa_supplicant

ctrl_interface_group=0

ap_scan=0

network={

    key_mgmt=IEEE8021X

    eapol_flags=0

    eap=TLS

    identity=`"<Common Name as specified in certificate>"`

    priority=6

    ca_cert="/setup/cacerts/scep/serverca.pem"

    client_cert="/setup/cacerts/scep/client.pem"

    private_key="/setup/cacerts/scep/client.key"

        }
```

   Add further entries according to your CA implementation.

   You can specify the certificate file and path in one of the following ways, for example if you access an external root certification authority. Note that only one file may be referenced via `ca_cert`. However, this file may contain several certificate entries.

```
ca_cert="/setup/cacerts/<root_extern>.pem"

ca_cert="/setup/cacerts/<subordinate_int>.pem"

ca_cert=`/setup/cacerts/<radius>`.ssl"
```

If the RADIUS certificate contains the NetBIOS name instead of the FQDN, you may use the following entry:

```
ca_cert=`/setup/cacerts/<root>`.pem
```

Important:

The spelling and case-sensitivity of the certificate file names must be identical to the names of the transferred certificate files.

2. If you want to use TPM 2.0 via WLAN, add the following engine parameters for the network:1 from eLux® RP 6 2103

```
network={

ssid="WLAN-ABC"

scan_ssid=1

key_mgmt=WPA-EAP

proto=WPA2

eap=TLS

engine=1            # Value must always be 1

engine_id="tpm2tss"    # Private Key is taken from TPM 2.0 module

dentity="__IDENTITY__"

priority=6
```

```
ca_cert="/setup/cacerts/scep/serverca.pem"

client_cert="/setup/cacerts/scep/client.pem"

private_key="/setup/cacerts/scep/client.key"    # Public part

      }
```

3. To transfer the `wpa.conf` file to the devices, use the Scout feature **Files configured for transfer**. Use the following destination:

LAN	`setup/scep/`
WLAN	`setup/wlan/`

For further information, see Advanced device configuration > Files in the **Scout** guide.

Use of multiple WiFi networks

- To use multiple SSIDs, set the **network** entry multiple times.

Example:

```
```

ctrl_interface=/var/run/wpa_supplicant

ctrl_interface_group=0

ap_scan=1

network=

{ ssid="" scan_ssid=1 key_mgmt=WPA-EAP eap=TLS identity="" priority=5
ca_cert="" ca_cert="" client_cert="" private_key="" }

network=

{ ssid="" scan_ssid=1 key_mgmt=WPA-EAP eap=TLS identity="" priority=6
ca_cert="" ca_cert="" client_cert="" private_key="" }
```
```

```
```

Use of variables

For identity and host name, you can alternatively set variables:

SPELLING	DESCRIPTION	OTHER
%IDENTITY%	Common name from certificate	For compatibility reasons, the legacy spelling before eLux RP 6.9.100 (variable name in uppercase letters and 2 x 2 underscores is still supported). Example: **IDENTITY**
%HOSTNAME%	Hostname from `terminal.ini`	

Variables may also be used for a part of a value. Prefixes and suffixes of a variable are pure strings that are passed through. Example: `identity="host/%HOSTNAME%"`

WPA configuration via template (SCEP)

– only if SCEP is used –

> **Note:**

> The following information is related to 802.1X configuration of LANs.

> If you use SCEP, you can benefit from the template file /setup/scep/wpa.conf.scep provided on the devices:

```
ctrl_interface=/var/run/wpa_supplicant

ctrl_interface_group=0

ap_scan=0

network={
```

```
    key_mgmt=IEEE8021X

    eapol_flags=0

    eap=TLS

    identity="%IDENTITY%"

    ca_cert="/setup/cacerts/scep/serverca.pem"

    client_cert="/setup/cacerts/scep/client.pem"

    private_key="/setup/cacerts/scep/client.key"

        }
```

Note:

The %IDENTITY%1 Former spelling: `__IDENTITY__` variable is replaced by the CN of the client certificate.

The client evaluates this template and creates the wpa.conf file if the following requirements are met:

- SCEP is configured in the terminal.ini file of the client. For further information, see Configuring SCEP for eLux in the SCEP guide.
- There is no individual /setup/scep/wpa.conf file available.

If required, modify the template and transfer it to the devices to /setup/scep/wpa.conf.scep. To do so, use the Scout feature **Files configured for transfer**. Note that if you have an individual /setup/scep/wpa.conf file,2 see Configuring WPA supplicant it has precedence.

Note:

The `wpa.conf` file generated from the template is created in a temporary directory and can only be viewed via the diagnostic files.

Diagnosis for 802.1X

Show client certificate in a shell

- Use the following command: `openssl x509 -in /setup/cacerts/scep/client.pem -noout -text`

All information about the certificate is displayed.

View log files

1. For the relevant OU, in the device configuration, enable enhanced logging.
2. Request the diagnostic files. For further information, see Retrieving diagnostic data in the **UniconOS Management Board** guide.

`/tmp/systemd-journal.log | Network log file`

Certificate handling with SCEP

Certificate handling with SCEP

SCEP is a protocol designed to simplify secure and scalable issuing of certificates to network devices in large-scale environments.

With SCEP, the devices take care of their certificates themselves. To do so, they require a one-time password created by an authorized person. With such a time-limited password (Windows: 60 minutes), a device can request a certificate from the SCEP service.

To prevent the certification authority (CA) from issuing arbitrary certificates with the one-time password, create a template in which you restrict the certificate classes.

By using SCEP you can simplify certificate issuing and management for uses such as

- 802.1X authentication for LAN and WLAN
- VPN connection via Cisco AnyConnect
- Connection via Citrix NetScaler® Gateway

The eLux® implementation is based on the **OpenSCEP** project. The following description requires the use of Windows NDES.

Windows as SCEP server

> **Note:**
>
> For Windows Server 2012 and later versions, NDES (Network Device Enrollment Service) is integrated in the Certification Authority (CA). The described procedure is to be seen as an example and can vary depending on the version and environment.

Installing NDES

1. Install the server role **AD CS** with the **Network Device Enrolment Service** feature .
2. Set up a service account for NDES which must be authorized on the CA later on.
3. Enter the information for the certificate from the Registration Authority (RA) to issue a signing certificate for this enrollment process.
4. Define the key length.

The new virtual directories are displayed in the IIS console below the default website.

Configuring templates

The SCEP server uses templates for submission to the CA.

1. Set the LDAP name of the template in the registry:

Default:	IPSecIntermediateOffline
EncryptionTemplate	`<templatename>`
GeneralPurposeTemplate	`<templatename>`
SignatureTemplate	`<templatename>`

2. Add the new template to the CA.
3. Configure the use of a one-time password in the registry (validity period, maximum number).

The one-time password allows a client that supports SCEP/NDES to contact the SCEP server directly and request a certificate. The client connects to the Windows SCEP server with the URL http://<CA_FQDN>/certsrv/mscep/mscep.dll

Configuring one-time password

Either

- disable passwords (option A): set to 0

or

- define fixed password (option B):

Permissions for NDES service account

- Set the permission to full control.

Modify SCEP application pool defaults

- Set **Load User Profile** option from false to true.

Requesting one-time passwords

1. Sign in to the SCEP Administrator website using your NDES service account: http://<CA_FQDN>/certsrv/mscep_admin/
2. Copy the challenge password to the configuration file scep.ini

Configuring SCEP for eLux®

How to enable eLux devices to use SCEP for certificate deployment:

> **Note:**
>
> The eLux package **Network Access Control** and the included feature package **SCEP** must be installed on the devices. This may require changing the image in **Software** in UniconOS Management Board, see Changing an image.

1. To configure the SCEP agent, use the scep.ini file.Then select the Scout feature Files configured for transfer to transfer the configuration file to the devices.For further information, see Configuring SCEP ini file.
2. Ensure that the devices are running on the correct time.We recommend that you configure a time server.

   The local time must match the time of the certificate authority.
3. To activate the SCEP agent, add the following entry to the terminal.ini file on the devices.To do so, use the Scout feature **Advanced file entries**:

File	/setup/terminal.ini
Section	Network
Entry	SCEP
Value	true

Please note case sensitivity. For further information, see Advanced file entriesin the **Scout** guide.

4. To use SCEP for 802.1X authentication for LAN, open the device configuration of the relevant OU in UniconOS Management Board. In the **Network** section, under **LAN**, click **Edit LAN connection**. In the **IEEE 802.1X authentication** group, select **Activate authentication via IEEE 802.1X**, see IEEE 802.1X authentication in the **UniconOS Management Board** guide. For further information, see Configuring 802.1X for eLux in the **802.1X** guide.

   When using SCEP for other purposes, however, make sure that the above-mentioned option is not selected for the network profile **LAN**.

The client creates a WPA configuration file from the template `/setup/scep/wpa.conf.scep`*.However, if you have transferred an individual WPA configuration file to* `/setup/scep/wpa.conf`*, it will take precedence over the template.*

Certificates for SCEP

The SCEP agent stores the following certificates in /setup/cacerts/scep by default.

CLIENT.PEM	CLIENT CERTIFICATE	
client.key	Private key of client certificate	2048 or40961 from eLux RP 6 2204 RSA key may be stored TPM 2.0 chip
serverca.pem	Server CA certificate	
serverca.key	Server CA key	required renewal of certificates
serverca.sig	Server CA signature	required renewal of certificates

CLIENT.PEM	CLIENT CERTIFICATE	
serverra.pem	RADIUS server certificate	

Devices with TPM 2.0

The authentication procedures via WPA supplicant for wired Ethernet connections and via IEEE 802.1X for WLAN components consider both storage options for the SCEP private key – the file system and the TPM 2.0 chip.

Client certificates continue to be stored in the file system. The roll-out procedure, with which the devices initially receive the client certificates, is carried out exclusively via wired Ethernet connections. Existing certificates can be renewed both via wired Ethernet connections and via WLAN.

Also the private key file of the SCEP client certificate is stored in the file system, by default. It is located under /setup/cacerts/scep or in the directory you have defined in the scep.ini file. When you perform an eLux firmware update to a version with TPM 2.0 support, existing private key files of the SCEP client certificate remain stored in the file system.

Private key in TPM 2.0 chip

- To use the TPM 2.0 module for the private key, make the following Advanced file entry:

File	scep.ini	
Section	Certificate	
Entry	UseTPM2	
Value	true	Default is false.

This parameter ensures that the private key is generated for each TPM 2.0 device in its TPM module.2 from eLux RP 6 2101 It is stored there only and cannot be displayed or exported. The key never leaves the TPM chip of the device.

When the certificate is renewed, the private key is reused.

The SCEP agent creates a certificate signing request (CSR) via openssl with the public key of the TPM module. Each time a connection is set up to an 802.1X-secured port, the system checks for a

valid certificate. The certificate must be signed with the private key of the TPM chip. If the certificate does not exist or is invalid, the connecting process is rejected.

The client.key file remains in the file system but only contains information on the public key. The retained file name allows devices without a TPM to use the same certificate.

To delete the key from the TPM 2.0 chip, delete the TPM in the BIOS. We recommend that you first revoke certificates that are still valid.

Configuring the SCEP ini file

> **Note:**
>
> You can use the example file on the devices to configure the SCEP agent: /setup/scep/scep.ini.sample

1. Create the text file scep.ini. Then insert the sections and make the entries described below.
2. To transfer the configuration file scep.ini to the devices under /setup/scep/, use the Scout feature Files configured for transfer. For further information, see Advanced device configuration > Files in the **Scout** guide.

The certificate request is created from the scep.ini data. You can optionally extend the certificate request with additional attributes. For further information, see Extended SCEP certificate request.

Sections and entries for scep.ini

SECTION	ENTRY	DESCRIPTION	EXAMPLE
Admin	URI	SCEP server address (URI)	`URI=http://ca.w2k12.samplete c-01.com/certsrv/mscep/`
	PROXY	Proxy server (optional)	PROXY=proxy.samplletec-01.com:3800
	ReNew	Number of days before the certificate expires and the device requests a new certificate (time	ReNew=30 (default)

SECTION	ENTRY	DESCRIPTION	EXAMPLE
		span for renewal) From this day on, the client tries to renew the certificate. The value must be less than the validity period of the certificate.	
	ReNewCheckOnlyClient1 from eLux RP 6 2302.1000	Alternative option to ReNew: Unlike ReNew, only the client certificate is checked, but not CA and RA certificates	ReNew=30 (Standard)
	ExpireCheck	Time interval in days, how often the ReNew date is checked	ExpireCheck=1 (default)
	challengePassword	One-time password for the request Valid for 60 minutes once only The password is deleted after the certificates have been successfully transferred.	(A): challengePassword=12345 (B): challengePassword=<password requested via http://CA_FQDN/certsrv/mscep_admin/>
Certificate	CNTYPE	Type (autoip,ip,dns,autodns,dnsfqdn,email)	CNTYPE=email
		autoip - The IP address of the device is used as **CN**.	

		ip - The IP address you specify as **CN** in the scep.ini (see next option) is used.	
		dns - The name you specify as **CN** in the scep.ini (see next option) is used.	
		autodns - The host name specified in the terminal.ini is used as **CN**.	
		dnsfqdn - The host name specified in the terminal.ini with the domain name appended is used as **CN**.	
		email - The email address you specify as **CN** in the scep.ini (see next option) is used.	
	CN	Certificate/Name Is filled by the system for CNTYPE=autoip,autodns,dnsfqdn	`CN=userxxx@sampletec-01.com`
	OU	Certificate/Attribute (optional)	OU=TestLab
	ORGANIZATION		ORGANIZATION=SampleTec
	LOCALITY		LOCALITY=Karlsruhe

	STATE		STATE=BW
	OU1	Further attributes for up to 6 OUs (optional).The OUs specified here can be used for a certificate request	OU1=Testlab
	OU2		OU2=KA_QA
	OU3		OU3=KA_DEV
	OU4		
	OU5		
	OU6		
	COUNTRY	Certificate/Country	COUNTRY=DE
	KEYLEN	Key length (from certificate template) Allowed values: 2048 or 4096 An incorrect key length leads to the event log entry **Incorrect Challenge Password** If a 4096 key cannot be generated in the TPM (for TPM 2.0 and UseTPM2), as a fallback a 4k RSA key is generated in the file system.	KEYLEN=2048
	CertStore	The certificate store can be freely selected. This parameter allows you to specify a new directory that the SCEP agent creates on the device. If you specify the default path /setup/cacerts/	CertStore=/setup/cacerts/scep

		scep, the private key for the SCEP client certificate for TPM 2.0 devices is stored in the TPM 2.0 module.	

Additional information

- The storage location can be defined individually for each certificate file.

The three file storage parameters have higher priority than the CertStore parameter. The specified directories must exist. Client certificate files can be merged.

- Access rights can be defined for the private key (if in file system).
- Certificates can be checked for validity according to the certificate revocation list (CRL)2 from eLux® RP 6 2101

Certificate	CA_Path	Path and file name of the server CA certificate	CA_Path=/setup/cacerts/scep/serverca.pem (default)
	Client_Path	Path and file name of the client certificate	Client_Path=/setup/cacerts/scep/client.pem (default)
	ClientKey_Path	Path and file name of the private key file of the client certificate	ClientKey_Path=/setup/cacerts/scep/client.key (default)

Note:

Specify the same value for **Client_Path** and **ClientKey_Path**, for example `/setup/cacerts/scep/client.pem`, if you want to merge the two files.

	RA_Path	Path and file name of the RADIUS server certificate	/setup/cacerts/scep/serverra.pem (default)
	MODE	For additional use of a RADIUS server certificate: RA Allowed values: CA (default), RA	CA
	ClientKey_Permission	Access rights for the private key file,0400 - Can only be read with root rights (default), 0444 - Can also be read by users	ClientKey_Permission=0400
	UseTPM2	For devices with TPM 2.0 chip From eLux RP 6 2101: The private key is generated in the TPM 2.0 module and remains there. eLux RP 6.10 and 11: The private key file is stored in the TPM 2.0 chip. Allowed values: false (/en-us/unicon-elux-scout/2611/short_guides/scep/scep_config_elux/scep_config_certificate.html).	UseTPM2=true

	TPM2Fallback3 from eLux RP 6 2204	If the private key cannot be stored in the TPM 2.0 module (for example, on devices with other TPM versions), it is stored in the file system. Allowed values: true (default), false false: There is no fallback. On devices without TPM 2.0, the key is not stored at all.	TPM2Fallback=true
	CrlCheck4 from eLux RP 6 2101	Optionally, specify up to five certificates (in addition to CA_Path, Client_Path and RA_Path) you want the SCEP agent to check for validity according to CRLs. The check is done each time a connection is set up. If the connection remains active for longer, the check is carried out after the time interval you set under ExpireCheck.	CrlCheck1=/setup/cacerts/myrootca.pem CrlCheck2=/setup/cacerts/myintermediate.pem

	CrlCheckEnabled5 from eLux RP 6 2110	true (default): All certificates are checked for validity according to CRLs false: No revocation check is done.	CrlCheckEnabled=true
	FirstCrlOnly6 from eLux RP 6 2204	true: If multiple CRLs are registered, only the first one is downloaded and checked. This reduces the load on the PKI infrastructure. false (default): All existing CRLs are downloaded and checked.	FirstCrlOnly=true

Extended SCEP certificate request

Note:

Use this function to configure the SCEP agent with additional attributes that are not included with the scep.ini file.

The certificate request is created from the values in the scep.ini file and is available on the device in the file `/setup/scep/clientreq.conf.in.` This file is installed with SCEP as a template and contains the following entries by default:

```
[req]

prompt=no
```

```
distinguished_name=req_distinguished_name

string_mask=nombstr

attributes=req_attributes



[req_attributes]

challengePassword=__CHALLENGEPASSWORD__



[req_distinguished_name]

C=__COUNTRY__

ST=__STATE__

L=__LOCALITY__

O=__ORGANIZATION__

OU=__OU__

1.OU=__OU1__

2.OU=__OU2__

3.OU=__OU3__

4.OU=__OU4__

5.OU=__OU5__

6.OU=__OU6__

CN=__CN__



[__X509V3__]
```

```
subjectAltName=critical,__CNTYPE__:__ALTNAME__
```

The fields marked with underscores are parameters and are replaced by the relevant values in scep.ini.

If the attributes provided in the scep.ini are sufficient for your purposes, there is no need to edit the clientreq.conf.in file.

Extending certificate request with additional attributes

1. Retrieve the /setup/scep/clientreq.conf.in file installed with SCEP from a device, for example by using the Diagnostics feature.
2. Edit the file. Add any other openSSL sections and attributes. For further information, see `<https://www.openssl.org/docs/man1.0.2/man1/openssl-req.html>`
3. Transfer the file to the relevant devices under /setup/scep/clientreq.conf.in by using the Scout feature Files configured for transfer.

This file – not the scep.ini – is the source for the certificate request. Only delete fields that you do not need!.

Diagnosis for SCEP

Show client certificate in a shell

Use the following command:

```
openssl x509 -in /setup/cacerts/scep/client.pem -noout -text
```

All information about the certificate is displayed.

View log files

1. For the relevant OU, in the device configuration, enable enhanced logging.
2. Request the diagnostic files. For further information, see Retrieving diagnostic data in the **UniconOS Management Board** guide.

/tmp/systemd-journal.log	Network log file

/setup/logs/scepagent.log1 temporary version under /var/log/scepagent.log	Log file of the SCEP agent, contains last certificate transfer This file is not included in the template #System, but must be defined in an individual template. For further information, see Retrieving diagnostic data in the **UniconOS Management Board** guide.

Performing the migration

The migration from eLux® RP 6 to eLux 7 can be carried out easily via a firmware update. Below you will find details on the requirements and the procedure.

Prerequisites

- The migration enablement 1.6.3-1 package or any newer version is available in the UC_RP6_X64 container and is defined in the current eLux RP 6 image.
- The following migration files are available in the UC_ELUX7 container:

```
- `migrationfiles-<version>.tar.gz`

- `migration-system-<version>.idf`

- recovery_systemEFI.rdf
```

- The eLux 7 target image (.idf) is defined in the UC_ELUX7 container and has the same name as the current eLux RP 6 image (.idf).
- In the device configuration, the container variable `__CONTAINER__` is specified for the firmware path.

Preparing the migration

The migration enablement package for eLux RP 6 required for migration and the migration files for eLux 7 are provided via a separate download link.

Prepare both containers in **Software** in UniconOS Management Board:

1. Open **Software** > **Software library**, select the eLux RP 6 container UC_RP6_X64, and add the migration enablement package with **Add package**, see Adding packages in the **UniconOS Management Board** guide.

2. In the **Software** menu, click **Images**, and then open the current eLux RP 6 image (.idf) of the container UC_RP6_X64.
3. In the **Package library**, tick the migration enablement package. In **Defined packages**, click the package, and tick its feature for the One Step Migration.
4. Click **Solve dependencies**, and then click **Save image**, see Changing an image in the **UniconOS Management Board** guide.
5. Unzip the archive `migration-package-<version>.zip`, and copy the migration files (`migrationfiles-<version>.tar.gz`, `migration-system-<version>.idf`, `recovery_systemEFI.rdf`) into the folder of the container UC_ELUX7, see The container folders in the **UniconOS Management Board** guide.
6. In the **Software** menu, below **Images**, select the container UC_ELUX7, and create the eLux 7 target image (.idf) with **New image**, see Creating an image in the **UniconOS Management Board** guide.

    - The image should have the same name as the current eLux RP 6 image (.idf).
    - If a name other than the current eLux RP 6 image (.idf) is to be used for the eLux 7 target image (.idf), an Advanced file entry must be set before the migration:

File	Section	Key	Value
/setup/terminal.ini	UpdateInfo	MigrationIdfUrl	`<URL>`

    The URL must not contain the container variable `__CONTAINER__` in the firmware path. Instead, the container name must be specified. Example: `https://MyWebServ.domain.com/eluxng/UC_ELUX7/MyImage.idf`

> **Important:**

> The migration files must not be changed manually.

> The devices must not be switched off during the migration.

> Devices should be connected via LAN, WLAN, VPN and IEEE 802.1X during the migration. Using WWAN connections (i.e. LTE) is not supported. During the migration via VPN the VPN-connection needs to be reconnected after each reboot.

> Please do not use the firmware settings “Check for new image version on start / on shutdown” during the migration process.

> This can lead to an incorrect device configuration (e.g. Citrix® StoreFront™ application definition), which requires a remote factory reset.

Migration process

The migration is triggered by the installation of the migration enablement package with activated One Step Migration FPM on eLux RP 6, if the device fulfils the migration requirements for eLux 7.

The migration requirements to be checked are:

- 8 GB HDD or more
- 4 GB RAM or more
- Boot Mode UEFI
- eLux RP 6 versions: 2302 LTSR, 2209 CR, 2104 LTSR CU3, 2104 LTSR CU4

If the device does not meet the migration requirements, a new check is carried out each time the device is restarted as long as the migration enablement package is installed.

A system message informs the user that the migration is due in a few minutes. During the migration, an eLux 7 base image is installed first and the eLux 7 target image is installed automatically in a further step.

> **Note:**

> Once the migration is complete, it is no longer possible to downgrade to an eLux RP 6 version with the old partition layout using a firmware update. Please use PXE recovery or USB stick recovery if a downgrade is required.

Migration from Windows to eLux 7

Migration from Windows to eLux 7

Overview

The **Win2UniconOS.exe** migration tool enables automated migration of endpoints running **Windows 10** or **Windows 11** to **eLux 7**.

When you execute Win2UniconOS.exe, a new partition is created on your local hard drive. This partition serves as the starting point for the migration process, where the eLux 7 operating system, as defined by the image, is installed.

The migration can be executed:

- Locally or from a network location
- As part of automation tools like **Microsoft SCCM** to enable deployment of Win2UniconOS.exe and remote initialization

After migration, the device boots directly into eLux 7.

How the migration works

The migration script must be executed on the device that needs to be migrated, when the migration script is executed:

1. A temporary recovery partition is created on the local disk.
2. Migration files are copied to the partition.
3. The device restarts automatically.
4. eLux 7 is installed using the specified Image Definition File (IDF).
5. The device boots into eLux.
6. Optional onboarding to Scout can occur automatically.

Requirements

Device requirements

- Device must be running Windows 10 or Windows 11.
- Device must boot using **UEFI**.
- Legacy BIOS mode is not supported.
- Network connectivity must be available during migration.
- Sufficient disk space must exist for the recovery partition.
- Device must have at least 3328 MB of RAM (4 GB recommended).

Important:

- Devices not booted in UEFI mode cannot be migrated.
- Migration is supported whether Secure Boot or BitLocker are enabled or disabled.

Software requirements

1. Sign in to myelux.com.
2. Navigate to:

   Products & Downloads > Tools > Migration tools
3. Download the latest **Win2UniconOS.zip** package.

Perform the migration

Extract the migration tool

From the device to be migrated, extract the provided migration tool package to a known location on the computer:

Run the migration script

Open the Command Prompt as **Administrator** and run the tool with the required parameters.

Example 1: Migrate the Windows endpoint to the given eLux image and reboot automatically and begins installation.

Execute the script:

```
Win2UniconOS.exe URL=<IDF_URL>
```

Example:

```
Win2UniconOS.exe URL=https://repository.example.com/elux/image.idf
```

Example 2: Migrate the Windows endpoint to the given eLux image, write a log file, use a proxy and don’t reboot

Example:

> Win2UniconOS.exe URL=https://repository.example.com/elux/image.idf Log=migration.log Proxy=http://proxy.citrix.com Port=8888 Reboot=0

Win2UniconOS.exe command line reference

> Usage: Win2UniconOS.exe [URL=<URL>] [Proxy=<proxy>] [Port=<proxyport>] [ARCHIVE=<URL|localpath>] [OnlyCheck=<0|1>] [Reboot=<0|1>] [Debug=<0|1>] [Log=<logfile>]

PARAMETER	INFO	DESCRIPTION
URL	Required	Define to the relevant software container and the image definition file (IDF) which specifies the scope of the installation
Proxy		Defines a proxy server, e.g. `1.1.1.1`
Port		Defines the port of the proxy server, e.g. `8090`
ARCHIVE	URL or local path	Path to the local Win2UniconOS.zip ramdisk file or URL when hosting remote
Log		Defines a storage path for the log file, such as `C:\Temp\migration.log`. By default, the log file is saved in the same directory as the Win2UniconOS tool. Alternatively, you can use a network drive to ensure a persistent log file.
OnlyCheck	`0` or `1`	Perform a dry run without any modifications (default is 0). This parameter verifies if the

PARAMETER	INFO	DESCRIPTION
		requirements for UEFI, network, disk space are met and generates a log file.
Reboot	`0` or `1`	Define if the endpoint should reboot after the installation (default is `1`) If the reboot option is `0`, the migration will not start automatically, but on the next boot
Debug	`0` or `1`	Verbose installation messages are not displayed in the console output by default (set to `0`). The installation process stops in case of errors and provides error messages for different scenarios. If the parameter is not provided, the device will restart in case of an error without displaying an error message.

Advanced configuration using RAM disk

A RAM disk configuration can be used when additional configuration is required during migration, such as:

- IEEE 802.1X authentication to initiate the first connection to the network during the migration process.
- HTTPS repositories requiring certificate trust.
- Zero-touch onboarding to Scout Server and Scout Cloud Gateway.

The configuration is provided as a RAM disk archive that **must be named `8021conf.xz` and included inside the migration package**. This archive contains the required configuration for the migration process and is loaded into memory during execution for temporary use by the migration tool. The RAM disk is created as a cpio archive compressed with XZ and **must be generated on a**

Linux-compatible system, Windows Subsystem for Linux (WSL) can be used to create the RAM disk.

> **Tip:** You can create the directory structure for each configuration, by extracting the example archive `8021conf.xz` from the working folder using the following commands.
>
> ```
> mkdir conf && cd conf && xz -cd ../8021conf.xz | cpio -iv
> ```

> After executing the command above, the directory structure will be created under the `conf` folder.

Configuring 802.1X authentication for migration

This configuration allows the administrator to initiate device migration when connecting to a network that enforces 802.1X authentication. During the migration process, the device uses a generic certificate, defined in the `wpa.conf` file, to establish the initial network connection.

After the migration is completed, the device receives a configuration from the Scout Server that defines how it should connect going forward. The device then reconnects to the network using a device-specific certificate and configuration defined in `wpa.conf.elux`, enabling access to the Scout Server and retrieval of the initial configuration.

> **Note SCEP use cases:** If SCEP certificate support is required during migration to replace the generic certificate used in the migration process, the custom_scep_renew-1.1.0-1.UC_ELUX7-1.0.zip package must be included to enable the SCEP request.

> **Note:** Only include 802.1X configuration when required. Including it on a migration that will connect to non-802.1X networks will cause migration failure.

> Create the following directory structure with the required files:

```
.

├── etc

│   └── wpa_supplicant
```

```
|       ├── client.key
|       ├── client.pem
|       ├── serverca.key
|       ├── serverca.pem
|       ├── serverca.sig
|       ├── wpa.conf
|       └── wpa.conf.elux
```

- `certificate` files used by the migration tool to connect to the 802.1x network. The certificates used during and after migration **must have the same filename**.
- `wpa.conf` is used during migration process.
- `wpa.conf.elux` is applied after the migration to enable eLux to reach the Scout server.

Example configuration for `wpa.conf`:

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
ap_scan=0
network={
    key_mgmt=IEEE8021X
    eapol_flags=0
    eap=TLS
    identity="eLuxMigration"
    ca_cert="/etc/wpa_supplicant/serverca.pem"
    client_cert="/etc/wpa_supplicant/client.pem"
```

```
    private_key="/etc/wpa_supplicant/client.key"
    private_key_passwd="PASSWORD_HERE"
}
```

Example configuration for `wpa.conf.elux`:

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
ap_scan=0
network={
    key_mgmt=IEEE8021X
    eapol_flags=0
    eap=TLS
    identity="eLuxMigration"
    ca_cert="/setup/cacerts/scep/serverca.pem"
    client_cert="/setup/cacerts/scep/client.pem"
    private_key="/setup/cacerts/scep/client.key"
    private_key_passwd="PASSWORD_HERE"
}
```

After successfully migrate the device, the client and server certificates defined above will be deleted from the device.

Adding SSL certificates for migration

This configuration allows the migration tool to trust the web server hosting the IDF image used to prepare the device during migration. If the web server is not trusted, the migration process will fail when attempting to access the website to download the required packages.

To establish HTTPS trust during the migration process, create the following directory structure and add the web server's root certificate.

```
.
├── etc
│   └── ssl
│       └── certs
│           ├── sslrootcertificate.pem
```

Enabling zero-touch onboarding

This configuration automatically onboards the device to the Scout Server or Scout Cloud Gateway, eliminating the need for users to manually enter the server URL and token during the onboarding process.

Create the json file and save as `zero_touch_onboarding.json` under the following structure:

```
.
├── user
│   └── setup
│       └── elux
│           ├── zero_touch_onboarding.json
```

Example of `zero_touch_onboarding.json` file for onboarding through SCG:

```
{
  "scgUrl": "scg.example.com",
```

```
  "scgToken": "onboarding-token"

}
```

Example of `zero_touch_onboarding.json` file for onboarding through Scout Server:

```
{

  "scgUrl": "scout-server.example.com",

  "scgToken": ""

}
```

Note: This configuration applies only to onboarding initiated through the migration tool. After a factory reset, the device will not automatically trigger onboarding again.

Create the RAM disk configuration archive

Generate the RAM disk archive:

```
./initrd-xz.sh <conf-folder> 8021conf.xz
```

Example:

```
./initrd-xz.sh /tmp/migration_tool/conf 8021conf.xz
```

Tip: (Optional) In case you can check the content of the `8021conf.xz` file, execute the following command:

```
xz -dc 8021conf.xz
```

Add it to the migration package:

```
zip Win2UniconOS.zip 8021conf.xz
```

Running the migration script with RAM disk package

From the Windows device to be migrated, execute the script:

```
Win2UniconOS.exe URL=<IDF_URL> ARCHIVE=<Win2UniconOS.zip path or URL>
```

Example:

```
./Win2UniconOS.exe URL=https://repository.example.com/elux/image.idf
ARCHIVE=Win2UniconOS.zip
```

The device restarts automatically and begins installation.

Debugging

When calling the Win2UniconOS.exe to perform the migration, you can enable the debugging by adding the `DEBUG=1` parameter. In case the migration fails due to 802.1X issues, you will be dropped into a debug shell and check for the log file `/tmp/wpasupplicant.log` for more details and any other log file under `/tmp/*`.

```
Win2UniconOS.exe URL=<IDF_URL> Debug=1
```

Logs are written to:

```
/tmp/
```

Example log:

```
/tmp/wpasupplicant.log
```

Use debug logs to diagnose:

- Network authentication failures
- Download issues
- Migration interruptions

Result of migration

After successful migration:

- eLux replaces the existing operating system.
- Temporary migration files are removed, including certificates used for 802.1x authentication when configured.
- The device boots directly into eLux.
- The device can automatically onboard to Scout if configured.

Migration from IGEL OS 11 or 12 or Stratodesk NoTouch OS to UniconOS

Migration from IGEL OS 11 or 12 or Stratodesk NoTouch OS to UniconOS

Overview

The **igel2elux** migration tool enables automated migration of endpoints running **IGEL OS 11 or 12** or **Stratodesk NoTouch OS** to **UniconOS**.

The migration installs UniconOS directly on the device by creating a temporary recovery environment from which the operating system is deployed.

The migration can be executed:

- Centrally through **IGEL Universal Management Suite (UMS)**
- Through **Stratodesk NoTouch Center**
- Locally or from a network location

After migration, the device boots directly into UniconOS.

How the migration works

The migration script must be executed on the device that needs to be migrated, when the migration script is executed:

1. A temporary recovery partition is created on the local disk.
2. Migration files are copied to the partition.
3. The device restarts automatically.
4. UniconOS is installed using the specified Image Definition File (IDF).
5. The device boots into UniconOS.
6. Optional onboarding to Scout can occur automatically.

Requirements

Device requirements

- Device must boot using **UEFI**
- Legacy BIOS mode is not supported
- Network connectivity must be available during migration, either through a wired connection (LAN) or through Wi-Fi (see Migrating over Wi-Fi)
- Sufficient disk space must exist for the recovery partition

> **Important:**
>
> Devices not booted in UEFI mode cannot be migrated.

Software requirements

1. Open the Citrix Downloads page.
2. Download the latest **igel2elux.zip** package.

Perform the migration

Extract the migration tool

From the device to be migrated, unzip the provided migration tool package to a known location on the computer:

```
unzip igel2elux-<version>.zip -d /tmp/migration_tool
```

In the above example, /tmp/migration_tool is used as the working directory. The following steps in this article assume all commands are executed from this directory.

Set execute permissions for igel2elux.sh

Simply having the migration script on the device is not sufficient. The script must have executable permissions before it can be run.

After uploading the `igel2elux.sh` script to the device, set the correct permissions using the following command:

```
chmod +x igel2elux.sh
```

Run the migration script

Execute the script:

```
./igel2elux.sh --url <IDF_URL>
```

Example:

```
./igel2elux.sh --url https://repository.example.com/elux/image.idf
```

The device restarts automatically and begins installation.

Tip:

When executing the migration from **IGEL Universal Management Suite (UMS)**, ensure that **read, write, and execute permissions** are assigned to both the owner and others for the `igel2elux.sh` file.

igel2elux.sh command line reference

Usage: `igel2elux.sh [--url <URL>] [--proxy <proxy>] [--port <proxyport>] [--archive <URL | local path>] [--noreboot] [--wifi] [--log <logfile>] [--debug]`

PARAMETER	DESCRIPTION
`--url <URL>`	Define to the relevant software container and the image definition file (IDF) which specifies the scope of the installation
`--proxy <proxy>`	Defines a proxy server
`--port <port>`	Defines the port of the proxy server
`--log <logfile>`	Defines a storage path for the log file. By default, the log file is saved in the same directory as the win2elux tool.

PARAMETER	DESCRIPTION
`--noreboot`	Does not restart the system after installation of the recovery image and the boot loader.
`--debug`	writes debug information in `/tmp/`.
`--archive`	Path to igel2elux.zip (URL or local path)
`--wifi`	Migrates over a Wi-Fi connection (`wlan0`) instead of a wired Ethernet connection. See Migrating over Wi-Fi.

Parameter `--wifi`

Enables migration over a Wi-Fi connection (`wlan0`) instead of a wired Ethernet connection. The parameter adds the kernel option `elux.pxe-interface=wlan0` to the options passed to the UniconOS recovery system, so the recovery system uses `wlan0` instead of the wired interface after it boots.

Prerequisites:

- The device must have a supported Wi-Fi adapter.
- The Wi-Fi configuration must be shipped in the RAM disk archive `8021conf.xz` (see Advanced configuration using RAM disk).
- `--wifi` requires an `--archive` that contains this `8021conf.xz`.
- WPA-Personal (`WPA-PSK`) and WPA-Enterprise (802.1X, EAP-TLS) are supported.

Advanced configuration using RAM disk

A RAM disk configuration can be used when additional configuration is required during migration, such as:

- IEEE 802.1X authentication to initiate the first connection to the network during the migration process.
- HTTPS repositories requiring certificate trust.
- Zero-touch onboarding to Scout Server and Scout Cloud Gateway.

The configuration is provided as a RAM disk archive that **must be named `8021conf.xz` and included inside the migration package**. This archive contains the required configuration for the

migration process and is loaded into memory during execution for temporary use by the migration tool. The RAM disk is created as a cpio archive compressed with XZ and **must be generated on a Linux-compatible system**.

> **Tip:**

> You can create the directory structure for each configuration, by extracting the example archive `8021conf.xz` from the working folder using the following commands.

```
# mkdir conf && cd conf && xz -cd ../8021conf.xz | cpio -iv
```

After executing the command above, the directory structure will be created under the `conf` folder.

Configuring 802.1X authentication for migration

This configuration allows the administrator to initiate device migration when connecting to a network that enforces 802.1X authentication. During the migration process, the device uses a generic certificate, defined in the `wpa.conf` file, to establish the initial network connection.

After the migration is completed, the device receives a configuration from the Scout Server that defines how it should connect going forward. The device then reconnects to the network using a device-specific certificate and configuration defined in `wpa.conf.elux`, enabling access to the Scout Server and retrieval of the initial configuration.

> **Note SCEP use cases:**

> If SCEP certificate support is required during migration to replace the generic certificate used in the migration process, the `custom_scep_renew-1.1.0-1.UC_ELUX7-1.0.zip` package must be included to enable the SCEP request. **Note:**

> Only include 802.1X configuration when required. Including it on a migration that will connect to non-802.1X networks will cause migration failure.

> Create the following directory structure with the required files:

```
.
├── etc
│   └── wpa_supplicant
│       ├── client.key
│       ├── client.pem
│       ├── serverca.key
│       ├── serverca.pem
│       ├── serverca.sig
│       ├── wpa.conf
│       └── wpa.conf.elux
```

- `certificate` files used by the migration tool to connect to the 802.1x network. The certificates used during and after migration **must have the same filename**.
- `wpa.conf` is used during migration process.
- `wpa.conf.elux` is applied after the migration to enable eLux to reach the Scout server.

Example configuration for `wpa.conf`:

```
ctrl_interface=/var/run/wpa_supplicant

ctrl_interface_group=0

ap_scan=0

network={

    key_mgmt=IEEE8021X

    eapol_flags=0

    eap=TLS
```

```
    identity="eLuxMigration"
    ca_cert="/etc/wpa_supplicant/serverca.pem"
    client_cert="/etc/wpa_supplicant/client.pem"
    private_key="/etc/wpa_supplicant/client.key"
    private_key_passwd="PASSWORD_HERE"
}
```

Example configuration for `wpa.conf.elux`:

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
ap_scan=0
network={
    key_mgmt=IEEE8021X
    eapol_flags=0
    eap=TLS
    identity="eLuxMigration"
    ca_cert="/setup/cacerts/scep/serverca.pem"
    client_cert="/setup/cacerts/scep/client.pem"
    private_key="/setup/cacerts/scep/client.key"
    private_key_passwd="PASSWORD_HERE"
}
```

After successfully migrate the device, the client and server certificates defined above will be deleted from the device.

Adding SSL certificates for migration

This configuration allows the migration tool to trust the web server hosting the IDF image used to prepare the device during migration. If the web server is not trusted, the migration process will fail when attempting to access the website to download the required packages.

To establish HTTPS trust during the migration process, create the following directory structure and add the web server's root certificate.

```
.
├── etc
│   └── ssl
│       └── certs
│           ├── sslrootcertificate.pem
```

Enabling zero-touch onboarding

This configuration automatically onboards the device to the Scout Server or Scout Cloud Gateway, eliminating the need for users to manually enter the server URL and token during the onboarding process.

Create the json file and save as `zero_touch_onboarding.json` under the following structure:

```
.
├── usr
│   └── setup
│       └── elux
│           ├── zero_touch_onboarding.json
```

Example of `zero_touch_onboarding.json` file for onboarding through SCG:

```
{

  "scgUrl": "scg.example.com",

  "scgToken": "onboarding-token"

}
```

Example of `zero_touch_onboarding.json` file for onboarding through Scout Server:

```
{

  "scgUrl": "scout-server.example.com",

  "scgToken": ""

}
```

> **Note:**
>
> This configuration applies only to onboarding initiated through the migration tool. After a factory reset, the device will not automatically trigger onboarding again.

Create the RAM disk configuration archive

Generate the RAM disk archive:

```
./initrd-xz.sh <conf-folder> 8021conf.xz
```

Example:

```
./initrd-xz.sh /tmp/migration_tool/conf 8021conf.xz
```

> **Tip:** (Optional)

> In case you can check the content of the `8021conf.xz` file, execute the following command: `bash` `xz -dc 8021conf.xz`

> Add it to the migration package:

```
zip igel2elux.zip 8021conf.xz
```

Running the migration script with RAM disk package

Execute the script:

```
./igel2elux.sh --url <IDF_URL> --archive <igel2elux.zip path or URL>
```

Example:

```
./igel2elux.sh --url https://repository.example.com/elux/image.idf --
archive igel2elux.zip
```

The device restarts automatically and begins installation.

Migrating over Wi-Fi

Devices that cannot be cabled can be migrated over Wi-Fi without any wired connection. The Wi-Fi connection is established in the recovery system and carried over to the installed UniconOS, so the device rejoins the Wi-Fi network after the migration and during onboarding.

> **Note:**

> Wi-Fi migration works only with Wi-Fi adapters whose driver and firmware support is present in the UniconOS recovery system. Verify your hardware model before a large rollout.

Provide the Wi-Fi configuration

Add the following files to the RAM disk configuration (see Advanced configuration using RAM disk):

```
.
├── etc
│   └── wpa_supplicant
│       ├── wpa.conf
│       ├── wpa.conf.wifi
│       ├── serverca.pem    (WPA-Enterprise only)
│       ├── client.pem      (WPA-Enterprise only)
│       └── client.key      (WPA-Enterprise only)
```

- `wpa.conf` is used by the recovery system to connect to the Wi-Fi network **during the migration**. Certificate paths point to the RAM disk (`/etc/wpa_supplicant/...`).
- `wpa.conf.wifi` is stored on the migrated device as its Wi-Fi configuration (`/setup/wlan/wpa.conf`) and used **after the migration**, including onboarding. Certificate paths point to `/setup/cacerts/scep/...` .
- Certificate and key files (`*.pem` , `*.key`) next to `wpa.conf.wifi` are copied to `/setup/cacerts/scep` on the migrated device. Use the same file names in both configuration files.

Example `wpa.conf` and `wpa.conf.wifi` for WPA-Personal (WPA-PSK):

```
ap_scan=1

network={
    ssid="CompanyWiFi"
    psk="SecretPassword123"
    key_mgmt=WPA-PSK
}
```

Example `wpa.conf` for WPA-Enterprise (EAP-TLS), used during the migration:

```
ap_scan=1

network={

    ssid="CompanyWiFi-EAP"

    scan_ssid=1

    key_mgmt=WPA-EAP

    eap=TLS

    identity="<CN of the client certificate>"

    ca_cert="/etc/wpa_supplicant/serverca.pem"

    client_cert="/etc/wpa_supplicant/client.pem"

    private_key="/etc/wpa_supplicant/client.key"

}
```

Example `wpa.conf.wifi` for WPA-Enterprise (EAP-TLS), used after the migration:

```
ap_scan=1

network={

    ssid="CompanyWiFi-EAP"

    scan_ssid=1

    key_mgmt=WPA-EAP

    eap=TLS

    identity="<CN of the client certificate>"

    ca_cert="/setup/cacerts/scep/serverca.pem"

    client_cert="/setup/cacerts/scep/client.pem"
```

```
    private_key="/setup/cacerts/scep/client.key"

}
```

Create the RAM disk archive and the `igel2elux.zip` as described in Create the RAM disk configuration archive.

Run the migration over Wi-Fi

```
./igel2elux.sh --url <IDF_URL> --archive <igel2elux.zip path or URL> --
wifi
```

If Wi-Fi cannot be used to start the migration, for example because the Wi-Fi setup is disabled on the device, start the migration over LAN and add `--noreboot`. Then unplug the LAN cable and reboot the device. The migration continues over Wi-Fi.

Debugging

When calling the igel2elux script to perform the migration, you can enable the debugging by adding the `--debug` parameter. In case the migration fails due to 802.1X issues, you will be dropped into a debug shell and check for the log file `/tmp/wpasupplicant.log` for more details and any other log file under `/tmp/*`.

```
./igel2elux.sh --debug
```

Logs are written to:

```
/tmp/
```

Example log:

```
/tmp/wpasupplicant.log
```

Use debug logs to diagnose:

- Network authentication failures
- Download issues
- Migration interruptions

Result of migration

After successful migration:

- UniconOS replaces the existing operating system.
- Temporary migration files are removed, including certificates used for 802.1x authentication when configured.
- The device boots directly into UniconOS.
- The device can automatically onboard to Scout if configured.

Overview

UEFI/BIOS update

By using UniconOS Management Board, you can perform an update of the UEFI or legacy BIOS of your devices remotely. For UEFI devices, the UEFI Capsule Update procedure is supported.

Download the latest manufacturer-specific UEFI or legacy BIOS firmware data from the relevant platform (such as the LVFS portal) or the manufacturer's homepage and provide them on your web server.

Updating UEFI devices to a newer UEFI firmware works in analogy to a software update and also supports configurable control by end users (defer and cancel). In UniconOS Management Board, administrators specify the UEFI file under **UEFI File** in the Firmware section of the device configuration. Then they start the UEFI update with the **UEFI update** command. The UEFI update on the end points is performed via the standardized Capsule Firmware Update procedure.

Alternatively, for legacy BIOS devices, you have the option of performing a BIOS update via the respective tool of the corresponding manufacturer. As an administrator, you start the legacy BIOS update via a user-defined Scout command with the parameters relevant for the respective manufacturer tool. The legacy BIOS update on the end device is performed via the respective manufacturer tool. The required manufacturer tools are provided through the companies Fujitsu, HP and Dell. These tools are integrated by Unicon into an eLux software package and made available on our myelux.com portal.

The devices can be equipped with different hardware.

UEFI/BIOS update for devices with encrypted system partition

Updating the UEFI or BIOS of devices with an encrypted system partition can be done via the described procedures without any problems. The only exception is Fujitsu FUTRO devices,

regardless of whether they are UEFI devices or legacy BIOS devices. Any attempt to update the UEFI or BIOS of these devices, if encrypted, is therefore prevented by eLux® and documented in the log file.

UEFI/BIOS configuration

In addition to a UEFI/BIOS update, some devices allow for configuring their BIOS remotely and, for example, setting a new BIOS password. This option depends on the hardware model.

Reference BIOS

Some devices (Fujitsu, Dell/Wyse, HP) allow for copying and transfer of a reference BIOS to other devices of the same type. Via a reference BIOS, new manufacturer firmware and/or a reference configuration can be transferred to other devices.

Disclaimer

Please note that Unicon supplies manufacturer-specific BIOS tools, updates and data exclusively for the purpose of software packaging and simplified logistics of provision. These manufacturer-specific BIOS tools, updates and data are taken over by Unicon from the respective manufacturer "as is", are not changed and passed on to you free of charge. This means that Unicon neither has influence on the creation nor knowledge of the concrete contents and functionalities of such manufacturer-specific BIOS tools, updates and data and therefore neither assumes responsibility for their individual suitability and/or usability nor responsibility for the consequences of implementing such manufacturer-specific BIOS tools, updates and data on your systems. Any liability and/or warranty of Unicon for such manufacturer-specific BIOS tools, updates and data shall therefore be expressly excluded, unless warranty or liability claims result from Unicon's own willful or fraudulent conduct in the context of passing on the BIOS tools, updates and data or from any other mandatory, non-negotiable, statutory reason for liability.

Requirements

Supported devices / UEFI

Dell	Wyse 3040, 5070, 5470 Latitude 3410, 3330, 3420, 3440, 5420, 5430, 5520, 5530, 5540 Optiplex 3000, 3090 Ultra, 5060 Precision 3260, WD19

Fujitsu	S 540, S 5010, S 5011, S 740, S 7010, S 7011, S 940, S 9010, S 9011 Q 940, LIFEBOOK E 5512, E 5410, E 5412, E 5511
Hewlett Packard	USB-C Dock G5 Device Update, Thunderbolt G4
Lenovo	ThinkCentre M75n, M625q ThinkPad X260, L14 Gen 1, L14 AMD, L14 Gen4

For a list of the devices that support the UEFI Capsule Update standard, refer to the **Linux Vendor Firmware Service** (LVFS): `<https://fwupd.org/lvfs/devicelist>`

This platform also serves as an online repository to which hardware manufacturers upload new firmware packed in .cab archives.

If UEFI Capsule Update is not yet supported by the vendor and the UEFI device is not listed via LVFS, use the procedure for legacy BIOS devices, provided a corresponding manufacturer tool is available.

Supported devices / legacy BIOS

Dell	Wyse 5020
Fujitsu	S 540, S 740, S 940, S 5011, S 7011, S 9010
Hewlett Packard	mt21, mt45, mt645, t430, t530, t610, t640, t740

Requirements for performing a UEFI/BIOS update via Scout

- eLux® RP 6
- Scout Enterprise Management Suite® including UniconOS Management Board with Software
- Web server (IIS) for UEFI/BIOS firmware of the hardware vendors

> **Important:**

> Do not perform the updates described, unless you have extensive experience with our software solutions Scout and eLux.

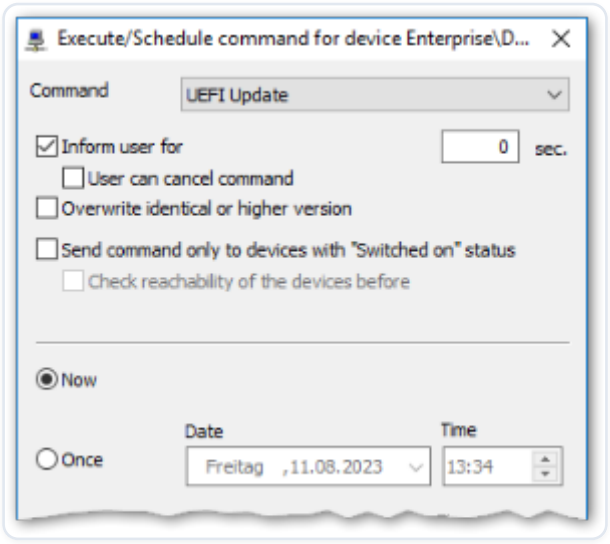
UEFI update in analogy to firmware update

– from Scout 15 2107 and eLux® RP 6 2107 –

For UEFI devices, you can conveniently update the UEFI firmware via a dedicated Scout command. The procedure is the same as for firmware updates (software): In a *.udf file you define the assignment of vendor-specific UEFI firmware to the device model types used. This UEFI file and the vendor-specific UEFI firmware archives are located in the eLux container. Finally, the UEFI file is referenced in the devices' device configuration. This allows UEFI devices that receive a UEFI update command to update to the appropriate UEFI firmware.

UEFI updates can be applied to individual devices, OUs, Dynamic Device Groups or devices selected in the **All devices** window. Administrators may schedule a UEFI update using the command dialog or create a UEFI update notification that will be evaluated on the next connection to the devices.

The options defined in the device configuration under **Firmware > Reminder settings** for users to defer an update are applied to both, firmware updates (software) and UEFI updates. Note that as with firmware updates, the administrator needs to additionally enable the **Inform user** option for each update command.



Since the UEFI update procedure and firmware update procedure (software) are structured similarly, the same features apply for both procedures:

- Initialization via Scout command or notification
- Update start can be controlled by users with appropriate configuration (**Firmware > Reminder settings** and user options in command dialog / notification).

- In the Scout Console, the update status is displayed for each device in its **Properties** window. Click to define which fields you want to show.
- Each update process is logged. For further information, see Command results and update information in the Scout guide.
- If an update fails, no system-side efforts will be made to retry.

> **Note:**
>
> The UEFI of Fujitsu devices with an encrypted system partition cannot be updated. Any attempt to do so will be stopped by eLux and documented in the log file. Whether a downgrade can be performed and to which version is specified by the manufacturer in the .cab file.

- Preparing UEFI devices
- Providing new UEFI firmware
- Configuring UEFI updates
- Performing UEFI updates via Scout command
- Performing UEFI updates via notification
- Logging of UEFI updates

Preparing UEFI devices

For updating UEFI systems, a special update daemon is required, which is provided by Unicon™ in an eLux® software package. This package needs to be integrated into the image of your devices.

1. From our myelux.com portal, under **Downloads > eLux Software Packages**, download the following software package:
   - **UEFI BIOS update daemon**
2. In UniconOS Management Board, open **Software** > **Software library**, select the container of your image, and then click **Add package**.
3. In the **Import package** dialog, click **Choose files**, select the downloaded files, and then click **Import package**.

   For further information, see Adding packages in the **UniconOS Management Board** guide.

4. In the **Software** menu, click **Images**, and then open the relevant image. In the **Package library**, tick the **UEFI BIOS update daemon** package.
5. Click **Solve dependencies**, and then click **Save image**.

   For further information, see Changing an image in the **UniconOS Management Board** guide.
6. For the relevant devices, perform an eLux firmware update to the updated image.

The eLux software package **UEFI BIOS update daemon** is installed on the devices.

Providing new UEFI firmware

The vendor-specific UEFI firmware data must be made available on your web server. The manufacturers provide the UEFI firmware of their devices on the LVFS platform in the form of .cab archives. The way the device types are listed is up to the individual vendors and is unfortunately not consistent.

For models that come with different chipsets, make sure to use the appropriate .cab archive for the UEFI update. Example: Dell Latitude 5530

The chipset installed in your model is displayed in the Scout Console in the device properties.

> **Note:**
>
> If you cannot find a specific UEFI device via LVFS, please contact the vendor. It is also possible that **UEFI capsule update** is not yet supported for this type of device.

1. For all relevant device types, download the latest UEFI firmware archives from the LVFS platform.

   To shorten the names of the .cab archives or make them more meaningful, you may modify the file names. Leave the file extension .cab.

   Example original name: `860fceb3052faa6955ff4f3dfe337ca1736e08515f5ac402a59c2af5b9bbf8d1-firmware.cab` Example short name: `Dell_Latitude_3330_1130.cab`
2. Store the archives on your web server inside the eLux® container.

   Alternatively, you may create a dedicated container for UEFI firmware. The corresponding path must then be specified in the UEFI file (*.udf).

3. Check the MIME type settings of your web server and add any missing entries:

.cab	application/vnd.ms-cab-compressed	default setting of IIS
.udf	text/plain	for UEFI updates via Scout **UEFI Update** command

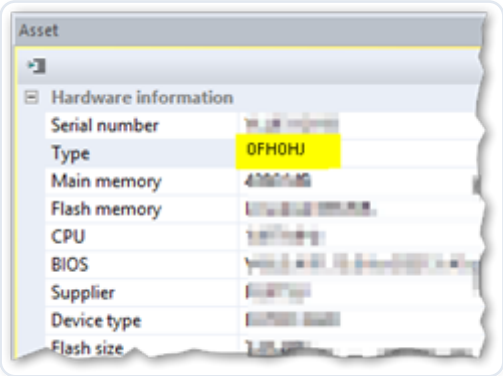
4. Follow the further steps in Configuring UEFI updates.

The new UEFI firmware is provided on the web server.

Configuring UEFI updates

The UEFI firmware archives need to be mapped to the relevant device types. This is done in a dedicated UEFI file which then is specified in the device configuration of all devices that are supposed to receive UEFI updates.

1. For all relevant device types, identify the type name. For a selected device, look up the Type name in the Scout Console, in the Asset window under **Type**.



2. As a **UEFI file**, create a text file with the following:

   - File name extension *.udf
   - Any file name

   For each model type, insert a line with `<type>=<firmware archive>`

   Example: 0FH0HJ=Dell_Latitude_3330_1130.cab1 original name of the firmware archive changed

uefi-123.udf

```
1  *<type>=<firmware archive>
2
3  OFH0HJ=Dell_Latitude_3330_1130.cab
4  ONP64J=Dell_Latitude_5420_TGL_1270.cab
```

If you have the UEFI firmware in a dedicated container, specify each model type with `<type>=<path/firmware archive>`

Example: 0FH0HJ=../UC_UEFI/dell_latitude_3330/Dell_Latitude_3330_1130.cab2 dedicated UC_UEFI container with different directories

3. Import your UEFI file (*.udf) into your eLux® container on the web server.
4. To modify the device configuration, for the relevant devices, open **Device configuration > Firmware**. Under **UEFI file**, enter the name of your *.udf file with the file extension.

1	UEFI file in the eLux container
2	From the specified data, the system generates a URL that is used by the devices to update their UEFI.
3	Users may postpone a UEFI update just like a firmware update if configured under Reminders. For further information, see Reminder in the **UniconOS Management Board** guide.

The UEFI file is located on your web server in the eLux container. In the device configuration of the relevant devices, the UEFI file is referenced.

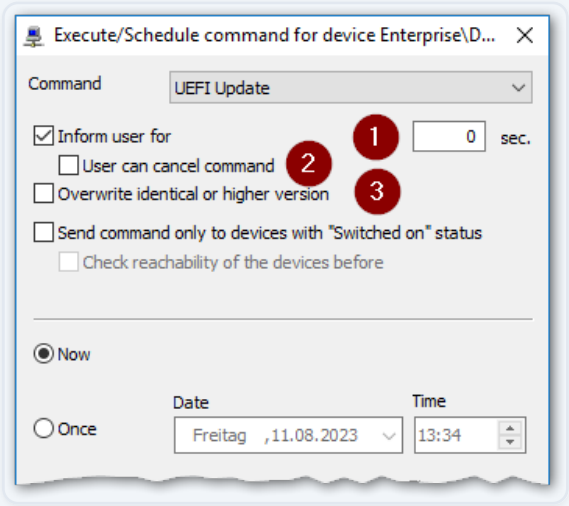
Note:

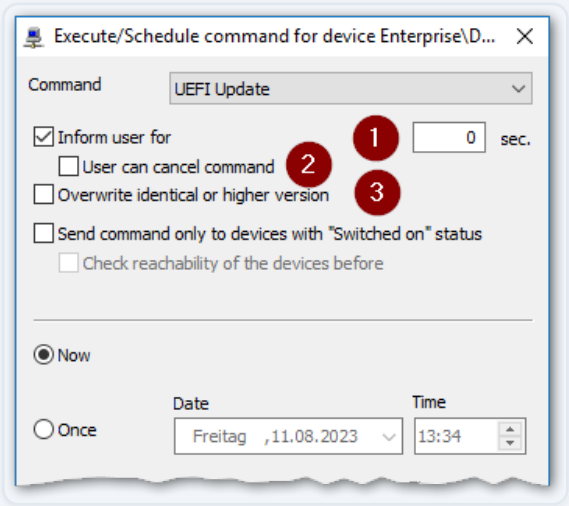
UEFI files can be globally predefined by the administrator just like image files. For further information, see Firmware preferences in the UniconOS Management Board guide.

Performing UEFI updates via Scout command

REQUIRES
• The relevant devices have the **UEFI BIOS update daemon** installed (/en-us/unicon-elux-scout/2611/short_guides/elux_bios_update/uefi_update_gui/uefi_epm.html).
• The UEFI firmware archives are available on the web server (/en-us/unicon-elux-scout/2611/short_guides/elux_bios_update/uefi_update_gui/uefi_binaries.html).
• The*.udf file is located in the eLux container and is referenced in the device configuration (/en-us/unicon-elux-scout/2611/short_guides/elux_bios_update/uefi_update_gui/uefi_update_configure.html).

1. Select a device, an OU, a Dynamic Device Group or devices in the **All devices** window. .
2. On the context menu, click **Commandos > UEFI update**.
3. Edit the following options:

	
1	Users will be informed before a UEFI update is performed. The system message displayed provides users with rescheduling options depending on the configuration (**Firmware > Reminder settings**). See also (User information before update) in the Scout guide. Optionally, specify a display duration of the system message in seconds. With 0

	
	seconds, the message will be shown until the user clicks one of the buttons.
2	Users are additionally allowed to cancel the command.
3	The UEFI system will be overwritten in any case. This option is mandatory for a downgrade.

4. Specify a time for the update process. For further information, see Commands and schedules in the UniconOS Management Board guide.
5. Click **Execute**.
6. Enter your device password.

The update process is triggered at the specified time. While the UEFI system is updated, for each device, the UEFI update in progress status is shown. In addition, detailed information about the current action is shown with a time stamp.

Performing UEFI updates via notification

– from Scout 15 2107 and eLux® RP 6 2107 –

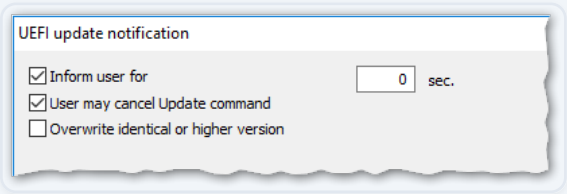
REQUIRES

- The relevant devices have the **UEFI BIOS update daemon** installed (/en-us/unicon-elux-scout/2611/short_guides/elux_bios_update/uefi_update_gui/uefi_epm.html).

REQUIRES
• The UEFI firmware archives are available on the web server (/en-us/unicon-elux-scout/2611/short_guides/elux_bios_update/uefi_update_gui/uefi_binaries.html).
• The*.udf file is located in the eLux container and is referenced in the device configuration (/en-us/unicon-elux-scout/2611/short_guides/elux_bios_update/uefi_update_gui/uefi_update_configure.html).

By using UEFI update notifications, you send an explicit one-time update request to selected devices that are evaluated with the next connection.

1. Select a device, an OU, a Dynamic Device Group or devices in the **All devices** window. .
2. On the context menu, click **Notifications > Initiate UEFI update**.
3. Edit the following options:

	
1	Users will be informed before a UEFI update is performed and are allowed to reschedule depending on the configuration.
2	Users are additionally allowed to cancel the command.
3	The UEFI system will be overwritten in any case. This option is mandatory for a downgrade.

   The options correspond to the dialog options for execution by command, see Performing UEFI updates via Scout command.

4. Confirm the notification and confirmation.

The notifications for UEFI updates are activated for the relevant devices.

For each device, in the **Properties** window, the UEFI update notification field shows the status Activated.

Deleting a UEFI update notification for one or more devices

UEFI update notifications can be deleted before the firmware has been updated:

- On the context menu, click **Notifications > Delete UEFI update notification**.

> **Note:**

> Notifications are always set or deleted for all selected devices regardless of whether they are only available for individual devices.

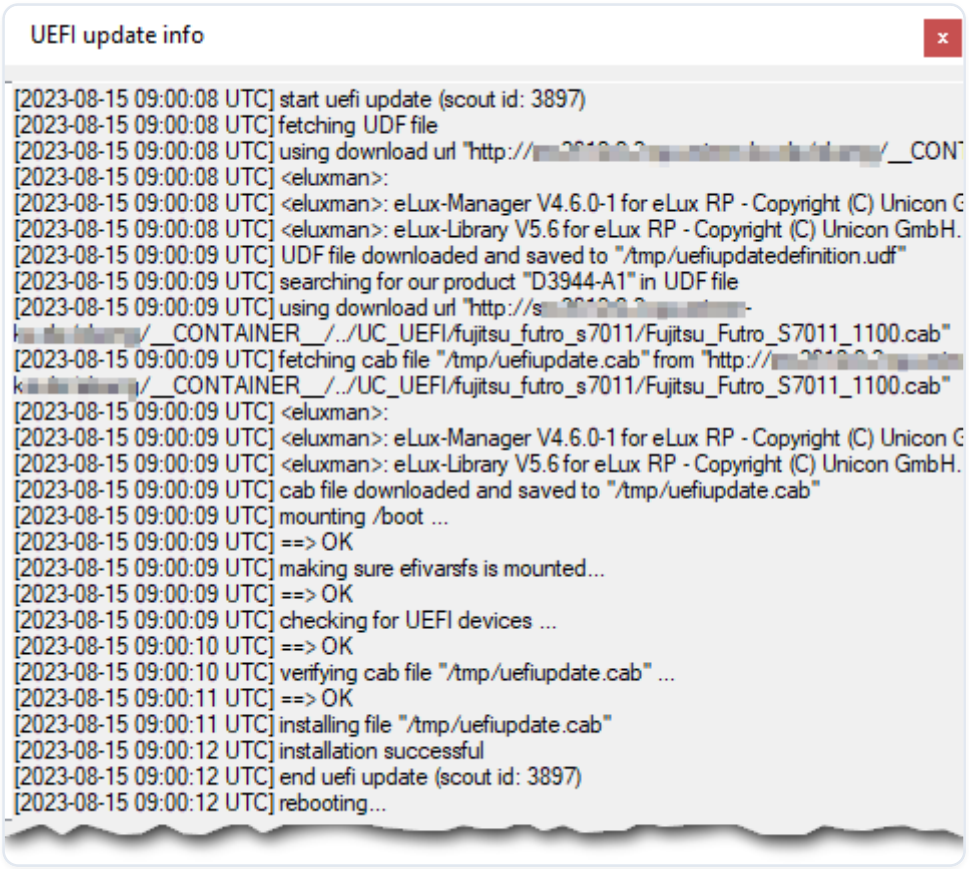
Logging of UEFI updates

UEFI updates can be triggered by a Scout command or by a Scout notification. For each device, in the Scout Console in the **Proerties** window, you may check the command status and other information:

1	Make sure that all relevant UEFI update fields are displayed. To show or hide fields, click the toolbar button.
2	Under **UEFI update**, you get relevant information about the update result and the time, size, provider, and image.
3	To view details, double-click **UEFI update state**.

Via the configured UEFI file (.udf), for each device type , a download-URL with the assigned UEFI firmware is used to transfer the appropriate .cab file. After a device restart, the installation of the UEFI firmware is started.

After the installation, the devices will restart again, this time using the new UEFI.



If it is a Fujitsu FUTRO device type and the system partition is encrypted, it will be found on the blacklist and the command will be canceled.

BIOS update for devices without UEFI

A BIOS update for legacy devices that are not yet equipped with UEFI is still possible. The update to newer BIOS firmware is initiated with the help of a user-defined command from the Scout Console and executed on the relevant devices.

We recommend to avoid downgrading to earlier firmware versions.

> **Note:**

> The BIOS of Fujitsu Futro devices with an encrypted system partition cannot be updated. Any attempt will be blocked by eLux® and documented in the log file.

- Preparing BIOS devices
- Providing new BIOS firmware
- Performing a BIOS update
- Different hardware models

Preparing BIOS devices

The manufacturer tools required for UEFI/BIOS updates are provided by Unicon in a special eLux software package. This package needs to be integrated into the image of your devices.

1. From our myelux.com portal, under **Downloads > eLux Software Packages**, download the following software package:

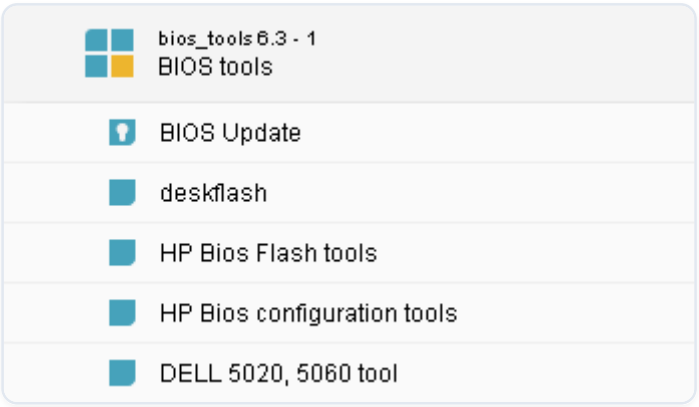
   - BIOS tools

2. In UniconOS Management Board, open **Software** > **Software library**, select the container of your image, and then click **Add package**.
3. In the **Import package** dialog, click **Choose files**, select the downloaded files, and then click **Import package**.

   For further information, see Adding packages in the **UniconOS Management Board** guide.

4. In the **Software** menu, click **Images**, and then open the relevant image. In the **Package library**, tick the BIOS tools package.
5. In **Defined packages**, click the BIOS tools package to show its features, and tick the feature for your hardware:

   - deskflash for Fujitsu
   - HP BIOS Flash for HP (BIOS update)
   - HP BIOS configuration tools for HP (BIOS configuration)
   - Dell tools for Dell

6. Click **Solve dependencies**, and then click **Save image**.

   For further information, see Changing an image in the **UniconOS Management Board** guide.

7. For the relevant devices, perform an eLux® firmware update to the updated image.

The eLux software package BIOS tools is installed on the devices.

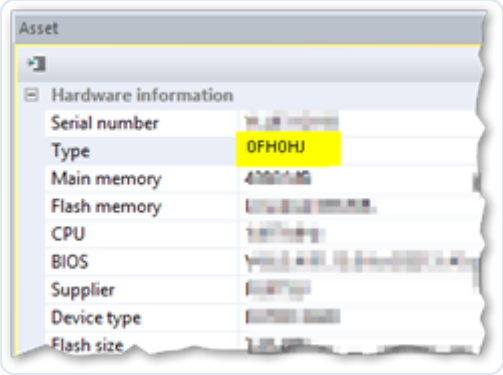
Providing new BIOS firmware

The vendor-specific BIOS firmware data must be made available on your web server.

1. For "legacy" BIOS devices, download up-to-date BIOS firmware from the vendor's homepage.

   The data are normally available as .bin or .bup archives.

2. On the web server, create a container named UC_BIOS
3. For each device type, under UC_BIOS, create a sub-directory. The name of the sub-directory must correspond exactly to the Type name of the devices. For a selected device, look up the **Type** name in the Scout Console, in the **Asset** window under **Type**.



> **Note:**
>
> It is important that the sub-directory names of the UC_BIOS container exactly match the type names in the Scout Console, in the Asset window (Hardware information). In the example, the name D3313-A1 represents the Fujitsu S920 device.

4. Copy the vendor-specific BIOS firmware archives into the sub-directories of the UC_BIOS container named according to the device types.

   Example: `eliasUC_BIOSD3313-A1`

5. Check the MIME type settings of your web server and add any missing entries:

.bin	application/octet-stream	default setting of IIS
.bup	text/plain	

> **Note:**

If you want to use individual names for the UEFI/BIOS files, we recommend following the 8.3 convention (especially for HP devices), which means a maximum of 8 characters for the file name and 3 characters for the file extension. For individual file extensions, add an appropriate MIME type.

The new BIOS firmware is provided on the web server.

Performing a BIOS update

REQUIRES
• On the devices, the eLux® software package **BIOS Tools** is installed
• On your web server, the UC_BIOS container with subdirectories and latest BIOS firmware archives is available Providing new BIOS firmware.

Important:

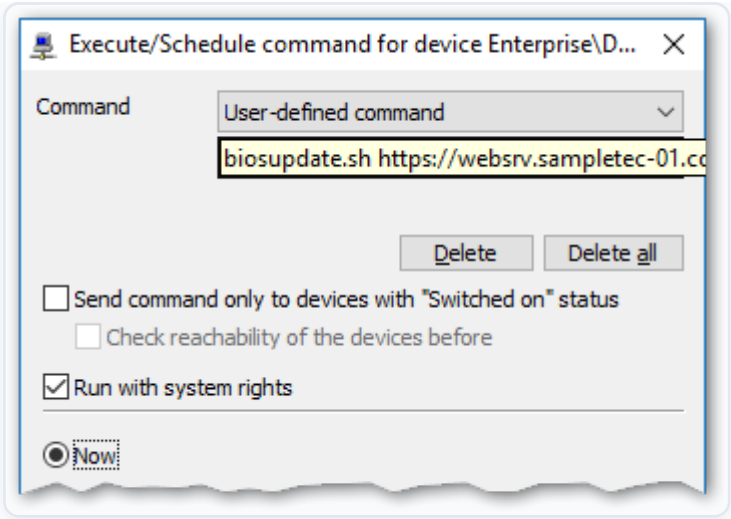
Avoid downgrading to earlier BIOS firmware versions.

1. For the relevant device, OU or Dynamic Client Group, open the context menu and click **Commands > User-defined command**.
2. In the command field, enter the script name biosupdate.sh and the URL to the manufacturer-specific container on the web server.

   If the BIOS is protected by a password, add the BIOS password. Use the following syntax: biosupdate.sh `<URL> <BIOS password>` Example: `biosupdate.sh https://websrv.sampletec-01.com/elias/UC_BIOS/__TYPE__/D3313-B1x.R1.15.0.bup <password>.`

   Note:

To update the BIOS of different device types via a common command, use the type parameter `__TYPE__`. For further information, see Different hardware models in the `__Scout__` guide.



The biosupdate.sh script starts the relevant manufacturer BIOS update tool.

3. Select **Run with system rights**.
4. Click `__Execute__`.
5. Enter your device password.

On the target devices, the BIOS update to the firmware data provided on the web server will be initiated.

Note:

You can predefine a user-defined command as a template: Create a predefined command and specify the password as a variable. The password is requested when the administrator executes the command. For further information, see Predefined commands in the **UniconOS Management Board** guide.

Result

The Scout Console triggers the transfer of the provided BIOS firmware data to the devices. The manufacturer tools driven by the script then install the new BIOS firmware on the devices. At this point, the devices already report the status successful: The initialization of the BIOS update was completed successfully. After the installation, the devices will be rebooted and then start with the new BIOS.

In the Scout Console, you may check the command status for each individual device in its **Properties** window

To view details, double-click **Command state**:

- The **Command info** dialog shows the individual steps performed for each command sent.
- The figure shows on the right side under **Command** the relevant user-defined command with exact syntax.
- Under **Result** it can be seen that this Fujitsu device type was found on the blacklist: For these devices the command will be aborted if the system partition is encrypted Otherwise the specified firmware archive is transferred.

Different hardware models

To have the UEFI/BIOS of different device types updated via a common command, use the type parameter `__TYPE__`. This is very helpful when you are dealing with a mixed hardware infrastructure.

The parameter is automatically resolved to the correct folder name of the UEFI/BIOS file.

> **Note:**
>
> The UC_BIOS container with correctly named subdirectories must be made available on the web server.

- In the user-defined command, following UC_BIOS, replace the directory name by the string `__TYPE__`. Example: `biosupdate.sh http://mywebserver/eluxng/UC_BIOS/__TYPE__/xxx.bin <password>`

The container parameter is resolved by the devices according to their hardware model, for example to D3313-A1.

Spelling of the container parameter

Make sure to use the correct spelling:

Two underscores followed by the string TYPE (all uppercase) followed by two more underscores: `__TYPE__`

Updating UEFI/BIOS configuration

For many devices, BIOS or UEFI configuration changes can also be made remotely with commands in UniconOS Management Board, see Commands and schedules.

- Changing individual UEFI/BIOS settings
- Exporting and changing a UEFI configuration /Fujitsu
- Using a BIOS as a reference /Fujitsu
- Updating a UEFI/BIOS configuration /HP

Changing individual UEFI/BIOS settings

For some devices, UEFI/BIOS configuration changes can also be performed remotely from UniconOS Management Board. This has been successfully tested for the following devices:

Fujitsu Futro S740, S940, S7010, S7011, S9010, S9011

REQUIRES
The eLux® software package BIOS Tools is installed on the devices.

> **Note:**
>
> To reference boot entries, check the manufacturer documentation for your devices to see which entries are available.

Setting/deleting the UEFI/BIOS password

1. For the relevant device, OU or Dynamic Client Group, open the context menu and click **Commands > User-defined command**.
2. Use the following commands: Replace current password with new password: `/opt/deskview/bin/biosset -PWD=<Old password> -NEWPWD=<New password>`

   Delete password: `/opt/deskview/bin/biosset -PWD=<Old password> -NEWPWD=`

   Set password, if not set: `/opt/deskview/bin/biosset -PWD= -NEWPWD=<New password>`
3. Select Run with system rights.
4. Click **Execute**.

Retrieving current boot order

- Use the following command:

  `/opt/deskview/bin/biosset -BOU –PWD=<password>`

For non-UEFI devices, the result is shown in the command info.

Changing boot order

REQUIRES
• The UEFI/BIOS CPU option VT-d must be disabled.

Use the following commands for **UEFI** devices:

`/opt/deskview/bin/biosset -BOU=<boot order> -PWD=<password>`

Example:

`/opt/deskview/bin/biosset -BOU=0x01-UEFI:FilePath0,0x0a-LAN0 -PWD=<password>`
`(first entry for HDD)`

- Use the following commands for **non-UEFI** devices:

  Changing boot order:

  `/opt/deskview/bin/biosset -BOOTORDER=1HDD,2LAN -PWD=<password>`

Disabling or enabling boot entry

REQUIRES
The UEFI/BIOS CPU option VT-d must be disabled.

- Use the following command to disable a boot entry:

  `/opt/deskview/bin/biosset -BOUD=<boot option> -PWD=<password>`

  Example for **UEFI** devices:

  `/opt/deskview/bin/biosset -BOUD=UEFI:FilePath0 -PWD=<password>`

  Example for non-UEFI devices:

`/opt/deskview/bin/biosset -BOUD=LAN -PWD=<password>`

- Use the following command to enable a boot entry:

  `/opt/deskview/bin/biosset -BOUE=<boot option> -PWD=<password>`

  Example for **UEFI** devices: `/opt/deskview/bin/biosset -BOUE=UEFI:FilePath0 -PWD=<password>`

  Example for non-UEFI devices: `/opt/deskview/bin/biosset -BOUE=LAN -PWD=<password>`

Configuring UEFI/BIOS with manufacturer tool

UEFI/BIOS configuration changes can be made for DELL devices, for example, via the relevant manufacturer tool. To do so, include the call of the tool in the user-defined command. Note that the entire parameter for the call must be set in quotation marks.

- Use a command according to the following example:

  `/usr/sbin/biosconfig.sh --tool "/opt/dell/dcc/cctk --ValSetupPwd=<password> -I" <URL>`

For the required parameters, refer to the manufacturer documentation, for example, the documentation for the DELL Client Command Suite. Note that the command is executed on the device without any further checks. The responsibility lies with the executing administrator.

> **Note:**
>
> You can predefine a user-defined command as a template: Create a predefined command and specify the password as a variable. The password is requested when the administrator executes the command. For further information, see Predefined commands in the **UniconOS Management Board** guide.

Exporting and changing a UEFI configuration /Fujitsu

– from eLux® RP 6 2204 –

The UEFI configuration can be exported as an .xml file for some devices. In the .xml file, you may change as many settings as you like before importing the file to the target devices. This has been successfully tested for the following devices:

Fujitsu Futro S540, S740, S940, S7010, S9010

REQUIRES

The eLux software package **BIOS Tools** and included feature packages **BIOS Update** and **Deskflash** must be installed on the devices.

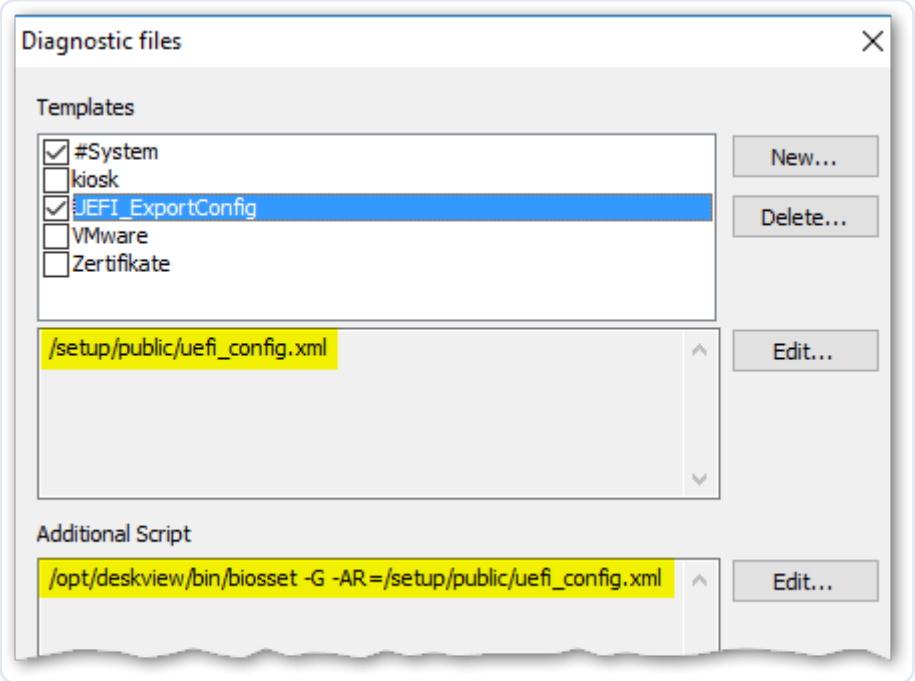
To use the **Diagnostics** feature, the object right **Edit diagnostic templates** is required (disabled by default)

1. To export the UEFI configuration of a device, use a user-defined command. We recommend that you execute the command using the **Diagnostics** feature. With it, you will transfer the .xml file in the same step.
2. Create a diagnostic template for which you specify the target file in the file list section.
3. In the **Additional script** section, create the actual command:

   ```
   /opt/deskview/bin/biosset -G -AR=/<path>/<targetfile.xml>
   ```

   Make sure to specify the .xml file extension for the target file. A password is not required.

   Example:



Note:

> As an alternative to the Diagnostics feature, you can execute the user-defined command directly on a device and get the target file by other means.

4. Change the desired settings in the .xml file using a text editor.

   For each entry, both the value **(SettingValue)** and descriptive text **(SettingValueText)** must be changed. For information on the settings and their possible values, see the Fujitsu User Manual of the **DeskView Client**, under **Expert mode**.

   Save the .xml file.

5. Place the .xml file on your web server below UC_BIOS in the relevant device type specific directory. For further information, see Providing new BIOS firmware.

6. To import the updated UEFI configuration, open the context menu of the target devices, for example, of a Dynamic Client Group, and choose **Commands >User-defined command**. Enter the following command:

```
`biosconfig.sh https://<server>/<path>/<sourcefile.xml> [UEFI password]`
```

   The password is required if the UEFI is password-protected.

   Example without password: `biosconfig.sh https://https://websrv.sampletec-01.com/eluxng/UC_BIOS/D3544-B1/BIOS_settings/uefi_config.xml`

> **Important:**
>
> The .xml file must have the filename.xml format.

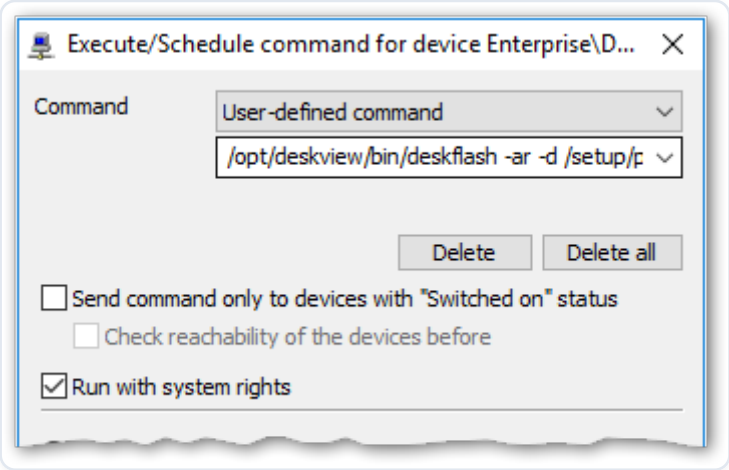
7. Check in the **Command info** or **Command history** whether the command has been executed successfully.

Using a BIOS as a reference /Fujitsu

As an alternative to updating the BIOS configuration via user-defined commands in multiple steps, you can copy the BIOS from a fully configured client and transfer it to other devices within the same model series.

> **Note:**
>
> The following procedure applies to Fujitsu devices within the same Futro model series.

1. For the configured device (reference client), open the context menu and click Commands > User-defined command.
2. Enter the following command: /opt/deskview/bin/deskflash -ar -d /setup/public
3. Select **Run with system rights**.



4. Click **Execute**.

   The BIOS is saved to a .bup file in the local directory /setup/public of your Futro client.
5. Copy the .bup file to a USB stick or use the **Diagnostics** feature to save it on the stick.
6. Make sure that the relevant vendor-specific directory exists within the UC_BIOS container on the web server. For further information, see Providing new BIOS firmware.
7. Copy the .bup file from the USB stick to the relevant subdirectory on the web server. Example: `UC_BIOS/D3313-A1 for device type D3313-A1`.
8. For your target devices, perform a BIOS update.For further information, see BIOS update for devices without UEFI

Updating a UEFI/BIOS configuration /HP

– currently available for the LTSR version only –

To update any UEFI/BIOS settings, configure the UEFI/BIOS of a reference system and then transfer it to the target devices of the same type.To only change the UEFI/BIOS password, you do not need a reference system and can start with step 6.

> **Note:**
>
> The following procedure applies to the HP devices t730 and t630.

1. For the configured device (reference client), open the context menu and click **Commands > User-defined command**.
2. Enter the following command: `/usr/AMI-HP630/LnxBCU.sh /Get:/setup/public/biossettings.txt`
3. Select **Run with system rights**.
4. Click **Execute**.
   - The UEFI/BIOS settings are saved to the text file biossettings.txt in the local directory /setup/public of your HP device.
5. Copy the `biossettings.txt` file to a USB stick or use the **Diagnostics** feature to save it on the stick.
6. Create the file `biossettings.zip.` If the UEFI/BIOS of your devices is secured by password or if you want to change the UEFI/BIOS password, you will need encrypted password binaries, see below.

   The `biossettings.zip` must contain the following files:

Update UEFI/BIOS settings on devices without UEFI/BIOS password	`biossettings.txt`
Update UEFI/BIOS settings on devices with UEFI/BIOS password	`biossettings.txt and curpassword.bin`
Update UEFI/BIOS settings and UEFI/BIOS password	`biossettings.txt, curpassword.bin and newpassword.bin`

Update UEFI/BIOS password only (You can skip the previous steps for this.)	`curpassword.bin and newpassword.bin`

7. Make sure that the relevant manufacturer-specific directory exists within the UC_BIOS container on the web server.For further information, see Providing new BIOS firmware.
8. Copy the biossettings.zip file from the USB stick to the relevant sub-directory on the web server.
9. For your target devices, perform a user-defined command (with system rights) to update the UEFI/BIOS configuration.To do so, use the following command: `biosconfig.sh https://<server>/<path>/biossettings.zip`

For further information, see Providing new BIOS firmware.

The UEFI/BIOS of your devices is updated with the UEFI/BIOS settings of the reference device.To check whether the command has run through without errors on all devices, use the Scout Console option **View > Command history**.

Creating password binaries

The passwords must be encrypted with the **HPQPswd.exe** tool supplied by HP for their UEFI/BIOS firmware.

1. Run **HPQPswd.exe** and then, in the password fields, enter the current password.
2. Save the encrypted binary file under the name curpassword.bin
3. Run **HPQPswd.exe** a second time and then, in the password fields, enter the new password.
4. Save the encrypted binary file under the name newpassword.bin

Analyzing UEFI/BIOS updates and configuration changes

We recommend that you check the UEFI/BIOS update and UEFI/BIOS configuration history immediately after performing an update or making changes to the UEFI/BIOS.

1. To open the command history, in the Scout Console, right-click the relevant device and click **Commands > Command info**.

The **Command info** dialog lists all commands executed on an individual device with their single steps.

2. After a UEFI/BIOS update or UEFI/BIOS configuration changes have been performed, restart the relevant devices with the new UEFI/BIOS.

Overview

Citrix UniconOS (formerly eLux 7) Dual Boot installs UniconOS next to an existing Windows installation on the same physical disk. A Windows installer shrinks the Windows system partition, creates a new partition for UniconOS, and writes and validates the UniconOS image. Windows remains fully intact and is the default operating system.

If Windows becomes unavailable — a security breach, malware infection, a failed update, or a system failure — the user can boot into UniconOS on the same device and quickly resume Citrix/VDI, browser-based, and remote-access work, with no dependency on Windows and no need for separate hardware.

The feature also provides:

- **Boot selection** — users choose Windows or UniconOS from a boot menu at startup, or an administrator sets a persistent default operating system.
- **Central management with Scout** — dual-boot devices onboard to Scout automatically on first boot and appear with a dedicated **DualBoot** device type.

As part of a business continuity plan, boot, update, and test UniconOS periodically so it is ready when it is needed.

What it is — and what it isn't. UniconOS Dual Boot is a *thin-client fallback* for business continuity, not a Windows recovery or data-access tool — it runs from its own partition and does not open Windows files, applications, or the Windows user profile. Reach your data the same way you would from any thin client — through a virtual desktop or published resources.

Prerequisites

Before installing Citrix UniconOS (formerly eLux 7) Dual Boot, ensure the following:

- You have **local administrator rights** on the device.
- The device runs **Windows 11** in **UEFI boot** mode. Legacy BIOS is not supported.
- The Windows system partition (**C:**) has enough free space for UniconOS plus a Windows reserve:

ITEM	SIZE
Minimum free space required (default)	36 GB
Reserved for Windows	20 GB
UniconOS partition, default size	16 GB
UniconOS partition, configurable range	16–64 GB

  If you choose a UniconOS partition larger than the 16 GB default, add the difference to the 20 GB Windows reserve. More free space is recommended.
- You have obtained:
  - The **UniconOS Dual Boot installer** (`UniconOSDualBootSetup.exe`).
  - The **UniconOS Dual Boot image** (`.zip` / `.dd`) for the matching UniconOS release.

  Contact your Citrix representative for access and guidance.

Disk encryption (BitLocker)

The installer detects the Windows system drive's BitLocker state automatically — no manual step is required. If BitLocker protection is active, the installer suspends it for the single install and resumes it automatically afterwards.

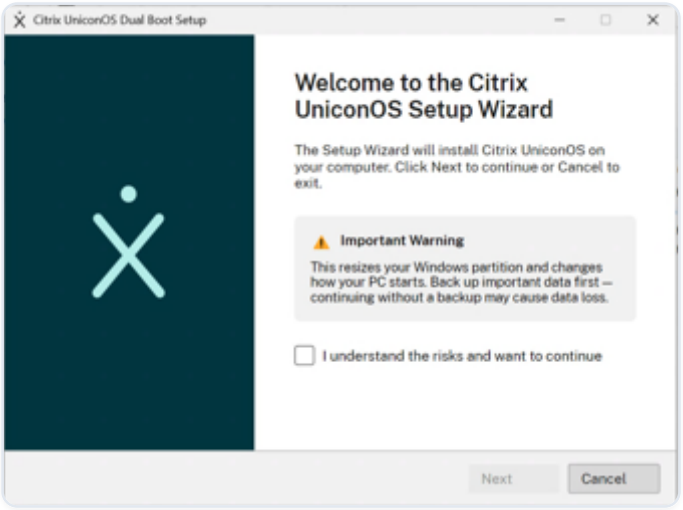
> **Note:**
>
> As a precaution, ensure the 48-digit BitLocker recovery key is escrowed before you roll out.

> Do **not** change the firmware Secure Boot setting while BitLocker is active without first escrowing the key. A Secure Boot toggle changes the measurements BitLocker validates and triggers a recovery-key prompt — this is a Windows/firmware behavior, independent of UniconOS.

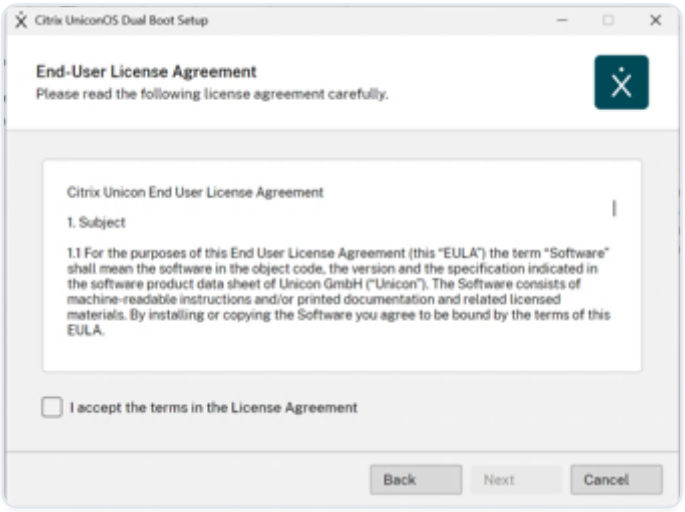
Installing

Install using the wizard

1. Save the installer and the UniconOS (formerly eLux 7) Dual Boot image file on the Windows device.
2. Double-click `UniconOSDualBootSetup.exe` and confirm the Windows prompt to allow the app to make changes. On the welcome screen, select **I understand the risks and want to continue**, then click **Next**.



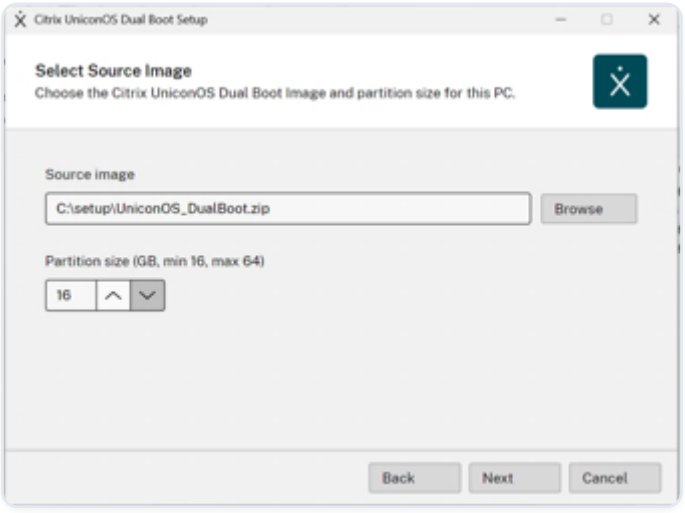
3. Read and accept the license agreement, then click **Next**.



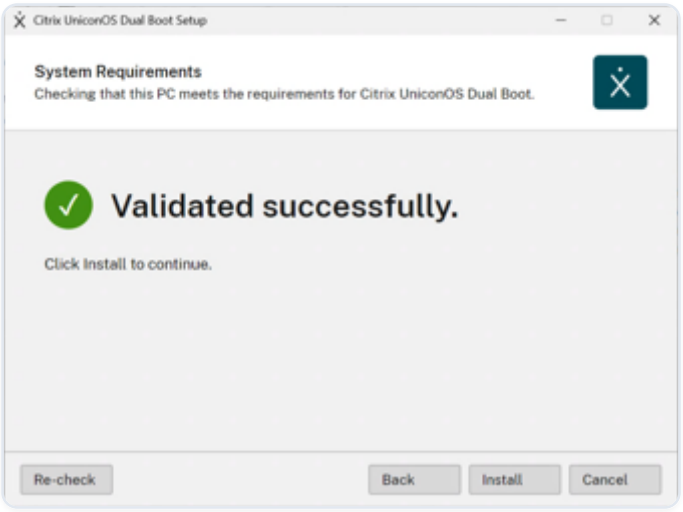
4. Provide the two installation details, then click **Next**:

   - **Source image** — click **Browse** and select the UniconOS Dual Boot image `.zip` / `.dd`.
   - **Partition size** — the UniconOS partition size in GB (minimum 16, maximum 64, default 16).

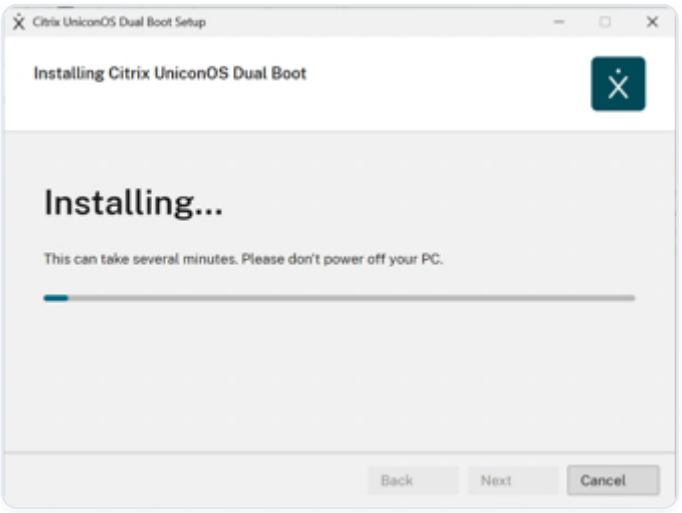
   The branded boot menu is shown by default, so no boot-menu option appears here. Scout onboarding is configured only via silent installation — see Managing dual-boot devices with Scout.



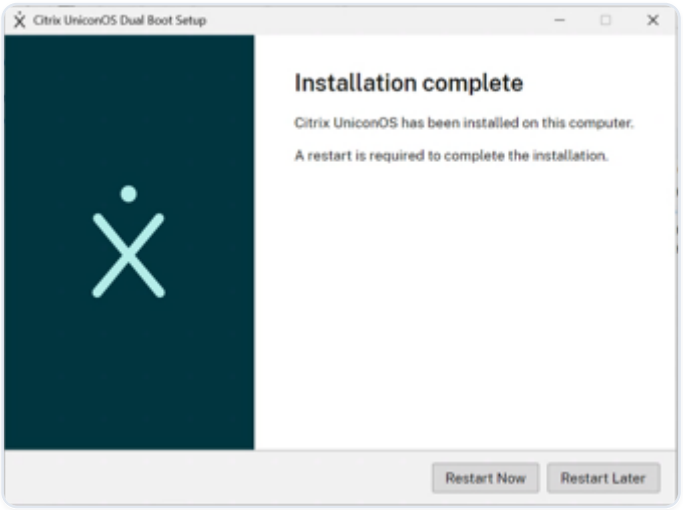
5. The installer verifies that the PC meets the system requirements. When it shows **Validated successfully**, click **Install**.



6. The installer carves the partition and writes the image.



7. When installation completes, click **Restart Now** to reboot immediately (a restart is required to finish) or **Restart Later**. After the restart, the device boots into the default operating system.



Install using the command line

To install unattended (silently), run the installer from the command line with the options and `NAME=value` properties below.

Options

OPTION	DESCRIPTION
`/quiet`	Silent, no user interface.
`/passive`	Non-interactive with a progress display; drives the MSI headless just like `/quiet`.
`/norestart`	Accepted for compatibility. A silent install never restarts automatically — it returns 3010 and leaves the restart to you.
`/log <file>`	Write a verbose log file.
`/uninstall`	Remove UniconOS Dual Boot.

Properties

PROPERTY	DEFAULT	DESCRIPTION
`IMAGE`	(required)	Path to the UniconOS image `.zip` / `.dd`.
`PARTITION_SIZE`	`16`	UniconOS partition size in GB (16–64).
`ACCEPT_EULA`	(not set)	Set to `1` to accept the license agreement — required for silent installation.
`DEFAULT_OS`	`windows`	Persistent default operating system: `uniconos` or `windows`.
`SETTINGS`	(none)	UniconOS settings file for automatic Scout onboarding — see Managing dual-boot devices with Scout.

PROPERTY	DEFAULT	DESCRIPTION
`BOOT_MODE`	`BOOTMENU`	Boot model: `BOOTMENU` shows the branded boot menu, `NOMENU` boots the default operating system directly.
`MENU_TIMEOUT`	(image default)	Seconds the boot menu is shown before it starts the default operating system.
`DISK`	`0`	Target physical disk number — set this on a machine with more than one disk.
`PROFILES`	(none)	Wi-Fi (WPA2-PSK) profiles file applied on first boot; used together with `SETTINGS`.
`TRAYAPP`	`0`	Set to `1` to install the system-tray boot-control app.

The branded boot menu is shown by default — no property is needed to enable it.

Only the properties listed above are passed to the installer. Any other `NAME=value` is ignored without an error, so a misspelt property silently installs the defaults — check spelling before a wide rollout.

Examples

```
UniconOSDualBootSetup.exe /quiet IMAGE="C:\temp\UniconOS_DualBoot.zip"
PARTITION_SIZE=32 ACCEPT_EULA=1
```

```
UniconOSDualBootSetup.exe /quiet IMAGE="C:\temp\UniconOS_DualBoot.zip"
ACCEPT_EULA=1 DEFAULT_OS=uniconos SETTINGS="C:\temp\terminal.ini"
```

To run the MSI directly with the same properties:

```
msiexec /i UniconOSDualBoot.msi /qn IMAGE="C:\temp\UniconOS_DualBoot.zip"
PARTITION_SIZE=32 ACCEPT_EULA=1
```

A restart is required to complete installation. In a silent deployment the installer returns **3010** to signal *restart required*; a failure returns **1603**.

Reinstalling or updating the image

To replace the UniconOS image on a device that already has Dual Boot installed, re-run the installer with a new `IMAGE` — no uninstall needed. The installer re-images the existing UniconOS partition in place, and runs silently or interactively:

```
UniconOSDualBootSetup.exe /quiet IMAGE="C:\temp\UniconOS_DualBoot.zip"
ACCEPT_EULA=1
```

The new image must fit the existing UniconOS partition. If it is larger, uninstall and reinstall instead. An interactive install offers a **Reinstall image** option when it detects an existing installation.

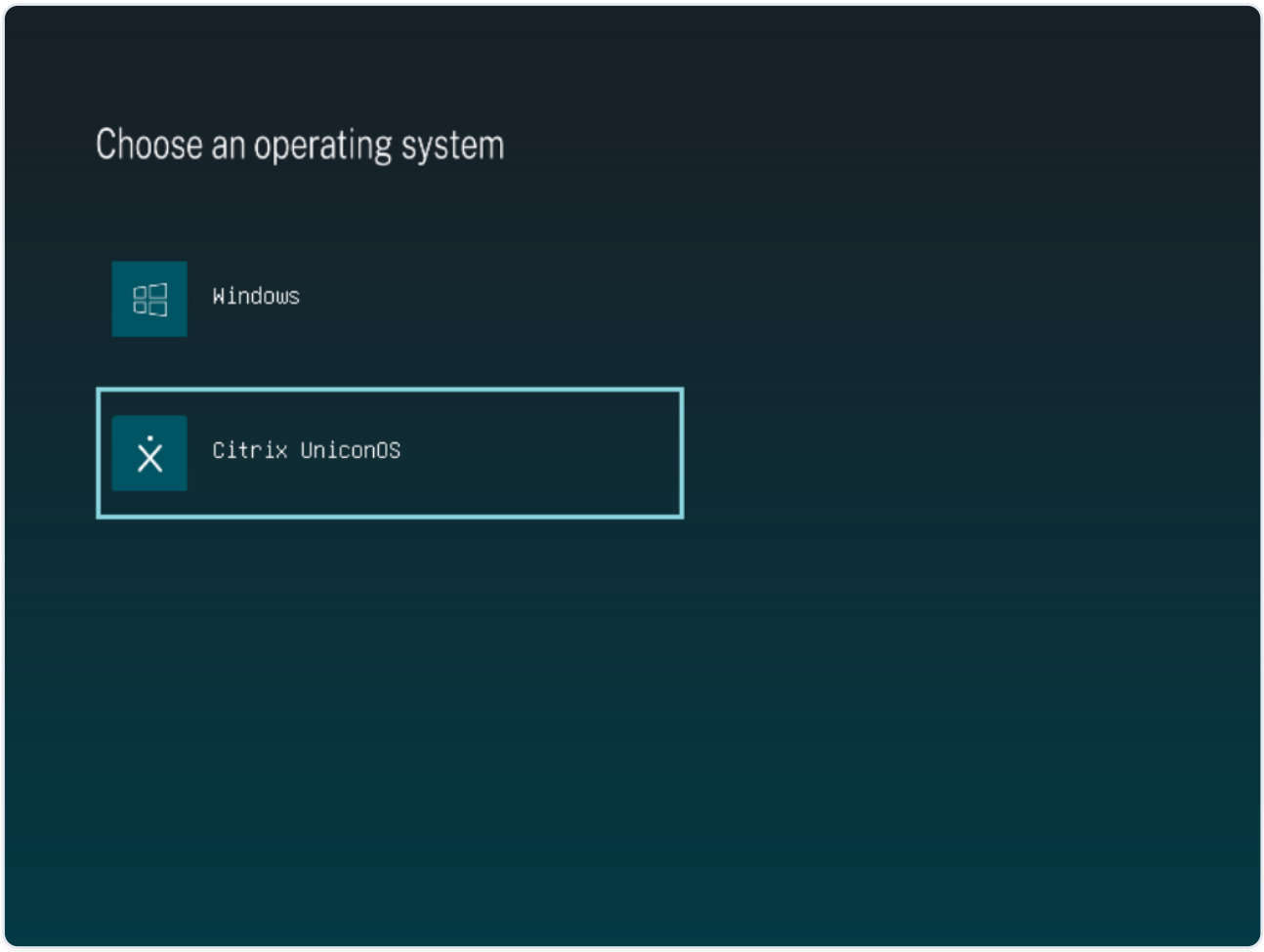
Selecting the Operating System at Boot

Using the boot menu

When the boot menu is enabled (the default), it appears for a short time at startup. The user selects the operating system to start:

- **Windows** — the default operating system.
- **Citrix UniconOS (formerly eLux 7)** — the UniconOS system installed alongside Windows.

If no selection is made within the menu timeout, the device starts the default operating system. This also applies in an emergency: if Windows is compromised, the user can boot UniconOS from the menu.



Setting the default operating system and boot-menu timeout

By default the device boots **Windows**. You can change the default operating system and the boot-menu timeout from Windows or from Scout.

From Windows

Run the installed command-line tool as administrator after installation.

To change the default operating system, use `--default`:

```
UniconOSDualBoot.exe boot-menu --default windows
```

```
UniconOSDualBoot.exe boot-menu --default uniconos
```

To also change the timeout, add `--timeout` and the number of seconds the boot menu is shown before the default operating system starts. For example, to make UniconOS the default and show the menu for 10 seconds:

```
UniconOSDualBoot.exe boot-menu --default uniconos --timeout 10
```

The new settings do not show in the boot menu right away. To apply them, start UniconOS once, for example by choosing it in the boot menu. After that, every restart uses the new default and timeout until you change them again.

The tool is installed at `%ProgramFiles%\UniconOSDualBoot\`.

From Scout

An administrator can also set the default operating system and the boot-menu timeout centrally, without touching the device in Windows. On a device that is already booted into UniconOS, run the following with **Run with system rights**:

```
/usr/sbin/dualboot-set-menu --default uniconos|windows --timeout <seconds>
```

The new settings apply the next time the device restarts.

Managing Dual-Boot Devices with Scout

Dual-boot devices connect to Scout the same way any other UniconOS (formerly eLux 7) device does, with a few dual-boot-specific details:

- **Silent install — zero-touch onboarding.** When you install silently and pass a settings file (`SETTINGS=`), the device onboards to Scout automatically on the first UniconOS boot, with no prompts. The Scout server, device name, and organization unit come from that settings file.

  > **Note:**
  >
  > Onboarding targets an on-premises Scout server only; the Scout Cloud Gateway (SCG) is not supported.

- **Interactive install — manual onboarding.** The wizard does not collect Scout details, so an interactively installed device does **not** onboard automatically. On its first UniconOS boot, complete onboarding on the device: enter the Scout server, select the organization unit, and finish.
- **DualBoot device type.** After onboarding, the device appears in Scout with a dedicated **DualBoot** device type, so dual-boot devices can be identified and targeted distinctly.

Logging

The installer and its tools write logs to `%ProgramData%\UniconOSDualBoot\logs` (`%ProgramData%` is a hidden folder). Use these logs to analyze problems during installation or boot management.

Recovery, Troubleshooting, and Uninstalling

If installation fails

All checks — UEFI/GPT, free space, and BitLocker state — run **before** any change to the disk.

- If a check fails, the installer stops (exit code **1603**) and the Windows partition is untouched.
- If a carve is cancelled or fails after it has started, the installer automatically rolls back the partial install and returns the disk to Windows-only (a user cancel reports exit code **1602**).
- Windows itself is never modified beyond the partition shrink, and its boot entry is left in place — **Windows always remains bootable**.

If power is lost during the image write, boot back into Windows and either re-run the installer or run `UniconOSDualBoot.exe uninstall` to clear the partial partition and reclaim the space.

Common issues

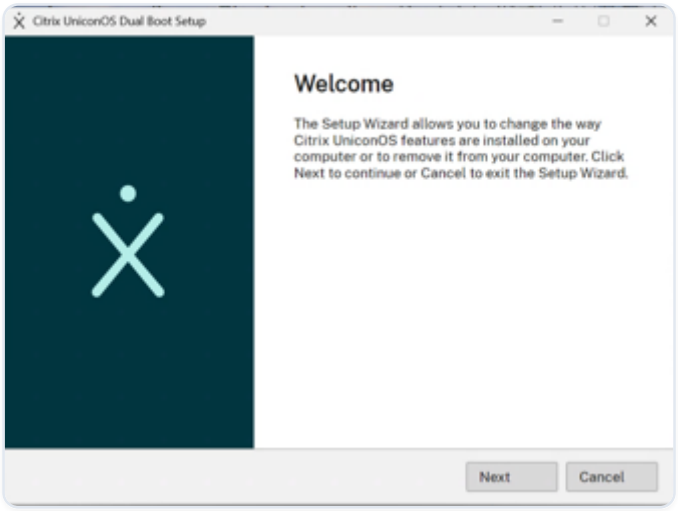
The table below covers the most frequent symptoms and how to resolve them:

SYMPTOM	CAUSE AND RESOLUTION
Install is disabled on the System Requirements page	Preflight failed a required gate. The host must be **UEFI + GPT** with enough free space (partition size + ~20 GB reserve). Secure Boot, TPM, and BitLocker are informational, not gates.
The branded boot entry might be missing after a Windows feature update	Expected — a startup task restores it automatically on the next Windows boot. No action needed.
A BitLocker recovery-key prompt appears	Not expected from installing UniconOS (formerly eLux 7). It usually follows a firmware Secure Boot change — enter the escrowed 48-digit key.

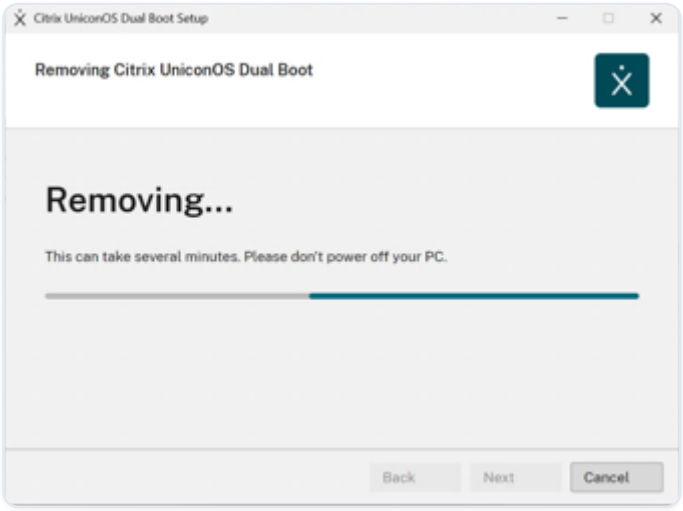
SYMPTOM	CAUSE AND RESOLUTION
The device did not onboard to Scout automatically	Automatic onboarding happens only on a **silent** install with `SETTINGS=`. For an interactive install, onboard manually on the first UniconOS boot — see Managing dual-boot devices with Scout.
Collecting logs for support	Gather `%ProgramData%\UniconOSDualBoot\logs`.

Uninstalling

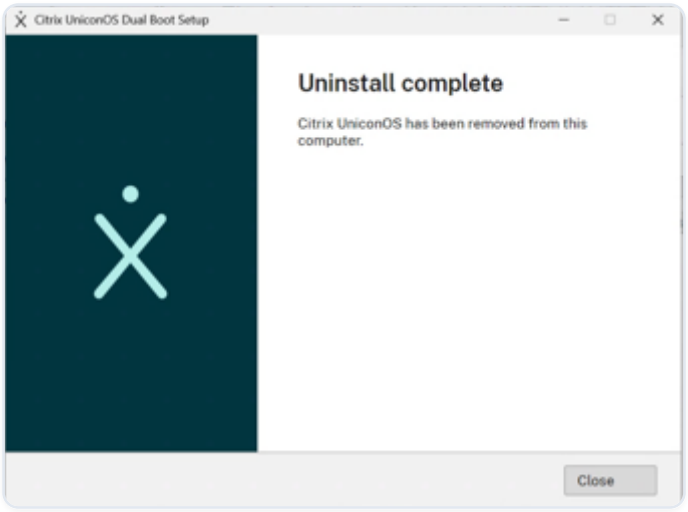
Uninstall UniconOS Dual Boot like any other program, or run the installer with `/uninstall`. On the welcome screen, click **Next**.



The device reverts to booting Windows only: the UniconOS partitions are deleted and the freed space is returned to the Windows partition.



When removal finishes, click **Close**.


Citrix UniconOS Dual Boot Setup
Uninstall complete
Citrix UniconOS has been removed from this computer.
Close

OEM Guides

Guides and integration documents with OEMs

Seamless Printing with ThinPrint

Printing is a common pain point in VDI and DaaS. End users expect it to "just work", while IT teams often face driver, network, and mapping issues. With Citrix Unicon OS (eLux) and ThinPrint Client, local and network printers are seamlessly available in Citrix sessions - without extra print servers.

The installation and configuration guide provides detailed information on ThinPrint.

Whitepapers

A whitepaper is a document that gives an overview on benefits, standards and technical topics, for example our licensing white paper.

Long Term Service Releases and Current Releases

Since January 2017, Unicon has been offering a Long Term Service Release (LTSR) of eLux RP and the management software Scout Enterprise Management Suite for download on our myelux.com portal.

By extending the life and support cycle of our software, Long Term Service Releases enable enterprises to retain a particular release for an extended period of time during which Unicon™ periodically publishes Cumulative Updates (CU). Cumulative Updates only include fixes for the LTSR components and features that help simplify the on-going maintenance. Opting for LTSR, customers obtain high-quality versions and can reduce update management efforts while they are sure to be provided with security fixes if necessary.

In parallel, Unicon carries on providing so called Current Releases (CR) for eLux RP and the Scout Enterprise Management Suite® on a quarterly basis. They are ideal for dynamic environments where the latest features and the most innovative technology are required.

For LTSR and CR product versions, consecutive version numbers are used. This means that a particular product version is either related to LTSR or to CR. Product versions will be labeled clearly to help customers distinguish LTSR from CR.

LTSR is included in the Scout Management license which must only be acquired once. Additional cost or new licensing is not required. Customers can choose between LTSR or Current Releases but they are also free to use both of them in a mixed environment with LTSR and CR versions for Scout Management Suite and/or eLux RP. Changing from CR to LTSR and vice versa is possible at any time. LTSR and CR versions are generally compatible unless explicitly stated otherwise in the release notes of the relevant product version.

Benefits of Long Term Service Releases

- Extended life and support cycle

- Simplified on-going maintenance and greater predictability
- Optimized management efforts through extended periods of time between product upgrades result in lower Total Cost of Ownership
- Additional fixes and security patches in Cumulative Updates

Differences between Current Releases and Long Term Service Releases

- Latest functionality and support of latest hardware platforms in CR
- Release cycle on a quarterly basis for CR
- New technology and fixes are combined in CR – Fixes are available with next CR

LTSR and CR are subject to different cycles. For further information, see Release cycles.

Long Term Service Releases and Current Releases

Since January 2017, Unicon has been offering a Long Term Service Release (LTSR) of eLux RP and the management software Scout Enterprise Management Suite for download on the Citrix Product Download portal.

By extending the life and support cycle of our software, Long Term Service Releases enable enterprises to retain a particular release for an extended period of time during which Unicon™ periodically publishes Cumulative Updates (CU). Cumulative Updates only include fixes for the LTSR components and features that help simplify the on-going maintenance. Opting for LTSR, customers obtain high-quality versions and can reduce update management efforts while they are sure to be provided with security fixes if necessary.

In parallel, Unicon carries on providing so called Current Releases (CR) for eLux RP and the Scout Enterprise Management Suite® on a quarterly basis. They are ideal for dynamic environments where the latest features and the most innovative technology are required.

For LTSR and CR product versions, consecutive version numbers are used. This means that a particular product version is either related to LTSR or to CR. Product versions will be labeled clearly to help customers distinguish LTSR from CR.

LTSR is included in the Scout Management license which must only be acquired once. Additional cost or new licensing is not required. Customers can choose between LTSR or Current Releases but they are also free to use both of them in a mixed environment with LTSR and CR versions for Scout Management Suite and/or eLux RP. Changing from CR to LTSR and vice versa is possible at any time. LTSR and CR versions are generally compatible unless explicitly stated otherwise in the release notes of the relevant product version.

Benefits of Long Term Service Releases

- Extended life and support cycle
- Simplified on-going maintenance and greater predictability
- Optimized management efforts through extended periods of time between product upgrades result in lower Total Cost of Ownership
- Additional fixes and security patches in Cumulative Updates

Differences between Current Releases and Long Term Service Releases

- Latest functionality and support of latest hardware platforms in CR
- Release cycle on a quarterly basis for CR
- New technology and fixes are combined in CR – Fixes are available with next CR

LTSR and CR are subject to different cycles. For further information, see Release cycles.

Release cycles

The table below compares the release schedules and related options. Note that the specified time periods can only provide general guidance. Unicon™ does not make product roadmaps public.

	LONG TERM SERVICE RELEASE (LTSR)	CURRENT RELEASE (CR)
Licensing	requires appropriate Citrix licensing	requires appropriate Citrix licensing
Access to Releases	Citrix Product Download portal	Citrix Product Download portal
Cumulative Updates (CU)	yes (CU typically only contain fixes for LTSR).	no (Rapid release cycles allow customers to add new features and all public fixes by upgrading to the next CR.)
Includes Fixes	yes (LTSR customers are required to update to next CU or next LTSR that contains the fix.)	yes (CR customers are required to upgrade to next CR that contains the fix.)
CU Schedule	4-9 months	n/a

	LONG TERM SERVICE RELEASE (LTSR)	CURRENT RELEASE (CR)
Release Schedule	12-18 months	2-3 months
Enhancements	No enhancements until next LTSR	Rapid release cycles, enhancements included

Unicon always supports the most recently published Long Term Service Release with the latest Cumulative Update, and the most recently published Current Release.

For further information, see Product and support lifecycle.

> **Note**

> UniconOS supports the parallel operation of LTSR and CR versions within a single managed endpoint environment.

Product and support lifecycle

> **Note:**

> The specified time frames are subject to change based on business needs or technical requirements.

> The Citrix products UniconOS and UniconOS Management pass through three life-cycle phases. Each phase is induced by a milestone.

The life-cycle of the products is driven by the customer requirements for new features and constantly evolving technologies including enhancements to existing products.

Milestones in the product lifecycle

General Availability (GA)	Date on which a major product release is made available for download on our technical portal. With GA, the maintenance period is started. Citrix customers receive continued support

	exclusively for the latest version of the Long Term Service Release (LTSR) provided on the Citrix Product Download portal with the latest Cumulative Update (CU), and for the latest version of the Current Release (CR) provided.
End of Maintenance (EOM)	After EOM, Citrix is no longer obliged to give software support. Citrix offers Extended Support on request (chargeable).
End of Life (EOL)	With EOL, no further security fixes are published. Support and downloads are no longer available for this product. EOL is a minimum of six months from the EOM date. After EOL, the customer may continue to use the product version within the product license terms. As the maintenance period has expired, the product can only be used on an as-is basis. The customer might be asked to update or migrate to the latest version.

Life-cycle milestones for individual products

OPERATING SYSTEM	GA	EOM	EOL
UniconOS (formerly eLux 7)	2024-05-29	2029-06-30	2029-12-31

ENDPOINT MANAGEMENT			
Unicon OS Management (formerly Scout 15)	2017-07-14	2029-06-30 (or upon GA of next major version)	n/a

SCOUT CLOUD GATEWAY			
Scout Cloud Gateway	2019-07-10	2027-06-30	n/a

The product life-cycle and the related services apply exclusively to already purchased products of current customers. They do not apply to other products or to any other third parties. Product lifecycles and support terms are subject to change if required.

Support terms

During the product life-cycle GA to EOM, services are provided as listed above and as stated in the EULA terms.

For EOM to EOL, Citrix offers additional support (Extended Support) on request. Extended Support is subject to a charge.

Compatibility client platform and Scout Enterprise Management Suite®

The following table shows the product versions required to manage devices through the Scout Enterprise Management Suite and to connect them via the Scout Cloud Gateway.

CLIENT-PLATFORM	SCOUT ENTERPRISE MANAGEMENT SUITE	SCOUT CLOUD GATEWAY
eLux 7 2404 and later	15 2309 / 15 2302 LTSR CU1 and later	
eLux RP 6.2 and later	15.2 and later	
Scout Agent for Windows platforms	15.2 and later	
eLux® 7 2404 and later		1 2209 and later
eLux RP 6.8 and later		1.1 and later

Legal notes

- Terms and Conditions of Use
- Privacy policy
- Cookie policy

Terms and Conditions of Use

Terms and Conditions of Use for

- `www.unicon.com`
- `uDocs.unicon.com`
- `myelux.unicon.com`

We are delighted that you have found our Unicon™ Internet portal (hereafter referred to as the “Unicon website”). We hope that you will find a product to suit you and take away plenty of ideas to help you design your commercial IT environment. Please take some time to read through these terms and conditions; if required, you can print them out by clicking on the relevant icon or save them to your own computer.

1. General

   Use of the Unicon website is permitted exclusively under the terms and conditions for use set out below. Unicon reserves the right to amend, supplement or replace these terms of use at any time. On logging in or, in the event that a separate login is not required, on start of use of the site, the user accepts the validity of these terms and conditions of use in their current version. Users use the Unicon website at their own risk.

2. Access to the Unicon Web Site

   Due to the nature of the Internet and computer systems, Unicon does not accept any liability for the uninterrupted availability of the Unicon website. Moreover, Unicon reserves the right to cease or limit the operation and/or provision of the website either in whole or in part at any time.

3. Use of the Unicon Website and the Content and Information Provided

All information published by Unicon is made available to users free of charge, unless otherwise specified. Each user may use and store (download) the information provided for his own personal needs. Any other use of the information, especially replication, modification or integration in publications or websites of any kind, is only permitted with prior consent from Unicon or the particular owner of the information. Unicon would like to point out that content and information may be subject to property rights; such property rights (in particular rights to the use of names and trademark rights) remain exclusively with Unicon or the relevant third-party owners.

All information provided by Unicon has been researched and compiled with the greatest care and to the best of its knowledge and belief, and, if originating from a third party, has been adopted without any alteration. However, it is the responsibility of every user to check that the information provided is correct, complete and up-to-date. Any Unicon liability due to incompleteness, incorrectness or outdatedness of the information concerned as well as any liability for damages resulting from any misinterpretation of the information provided by the user shall be limited to gross negligence or intent on the part of Unicon.

When using the Unicon website, the user shall also ensure that such use or any associated activity does not, under any circumstances:

- cause damage to persons, in particular minors, or infringe their personal rights;
- transgress standards of public decency;
- violate industrial property rights and copyright or other property rights;
- convey contents infected by viruses, so-called Trojan horses or other programs that can damage the software;
- enter, save or send hyperlinks or contents, which the user is not entitled to enter, save or send, particularly if these hyperlinks or contents violate obligations to maintain confidentiality, are unlawful or in breach of contract and are simply deliberately incorrect; and/or
- distribute advertising or unsolicited e-mails (so-called "spam") or inaccurate warnings of viruses, malfunctions or the like or incite persons to take part in competitions, snow-ball systems, chain letters, pyramid games and comparable activities.

Unicon shall not be liable for damages resulting from any misuse by the user of the information offered.

4. Links/Content Provided by Third Parties

The Unicon website may contain links and/or other references to websites of third parties. Unicon accepts neither any liability for the content of these third-party websites nor any ownership of these web-sites and/or their content, as Unicon does not have any influence over the linked content and information and does not monitor this content and information on a regular basis. Similarly, Unicon shall also not be liable for the quality, correctness and/or completeness of information from third parties where such information is explicitly indicated as such on the Unicon website. This also applies expressly to all information and content provided by Unicon sales partners on the Unicon website.

5. Software Downloads

   The use of any software offered by Unicon for download is governed by the applicable license conditions of the authorized supplier or the manufacturer concerned (license agreement). These terms and conditions are transferred with the appropriate software as a file or can be obtained from the manufacturer/authorized supplier of the software on request. As a rule, the software cannot be installed without the user's prior acceptance of the terms and conditions of the license agreement.

   The software offered is – unless otherwise specified – intended only for downloading and usage by end users and may only be used by end users. Any other use will be pursued according to criminal and civil law. Mandatory user rights shall remain unaffected within the particular scope of the law as applicable for the respective area of validity.

   Unicon accepts no liability for loss occurring directly or indirectly through the use of downloaded files.

6. Important Information About Computer Viruses

   Although Unicon makes every endeavor to keep the Unicon website virus-free, Unicon cannot provide any warranty or accept any liability for the Unicon website being virus-free. Before downloading information, software and documents, the user shall ensure that appropriate security precautions are taken and appropriate virus scanners used, both for his own protection and to prevent viruses infecting the Unicon website.

7. Rules for Use and Responsibility for Passwords

   The user is solely responsible for safeguarding any password issued for use of certain areas of the Unicon website against misuse. Should a user discover that his password has unintentionally been made accessible to any unauthorized persons or third parties or that there is a risk of misuse, the user is requested to inform Unicon of this immediately. Unicon will then take appropriate action to block the affected password. Notwithstanding this, however, Unicon shall not be liable for any damages resulting from any misuse of passwords, unless such

misuse was caused by gross negligence or willful intent on the part of Unicon or a vicarious agent of Unicon. Unicon shall also be entitled to revoke or restrict access authorization at any time by blocking the access data, without being required to cite reasons.

8. Limitation of Liability

   Unicon shall not be liable for any consequential damages, especially lost profits, production losses, interruption of operations or loss of data unless such liability is mandatory by law. As far as the liability can be excluded or limited according to law, but has not already been excluded or limited elsewhere within these terms and conditions of use, Unicon's liability to the user shall be limited to an aggregate maximum sum of €12,500. Mandatory legal liability exists in particular in the event of a grossly negligent violation of important obligations or willful conduct on the part of Unicon.

9. Court of Jurisdiction/Applicable Law

   Unicon does not accept any responsibility for the fact that information, software and/or documents from the Unicon website may also be accessed or downloaded at locations outside the country concerned, or that the downloading of such information, software and/or documents is permitted in these locations. Users who access the Unicon website from locations outside the country concerned are solely responsible for compliance with regulations that apply under the laws of that particular country. Accessing or downloading information, software and/or documents from the Unicon website from countries where such access or download is illegal is explicitly prohibited by Unicon.

   The laws of the Federal Republic of Germany are applicable. The court of jurisdiction is Munich as long as such place of jurisdiction can rightfully be agreed with the user.

10. Legal information

    © 2024 Unicon GmbH

    This document is copyrighted. All rights reserved.

    eLux® and Scout Enterprise Management Suite® are registered trademarks of Unicon GmbH in the European Union, GB and the United States.

    ScoutaaS® is a registered trademark of Unicon GmbH in the European Union, GB, the USA and Japan.

    All other product names are registered trademarks of their relevant owners.

Written Offer for eLux® and Scout products:

For software in binary form which is integrated in the product and which is licensed under an open source license that gives you the right to receive the source code for that binary, you can obtain a copy of the applicable source code from Unicon Entwicklungs- und Vertriebsgesellschaft mbH on physical media by sending a request by email to the following email address: `support@unicon.com`.

Your request by email must be sent within three (3) years of the date you received the product including the binary that is the subject of your request.

Your request should include:

- The name of the binary for which you are requesting the source code
- The name and version number of the product containing the binary
- The date you received the product
- Your name/company name and
- Your return mailing address and email.

Please note, that

Unicon GmbH may charge you a fee to cover the cost of physical media and processing.

Adobe, Acrobat Reader and PostScript are registered trademarks of Adobe Systems Incorporated in the United States and/or in other countries.

Broadcom is a registered trademark of Broadcom Corporation in the U.S. and/or in other countries.

CardOS is a registered trademark and CONNECT2AIR is a trademark of Siemens AG in Germany and/or in other countries.

CUPS and the Common UNIX Printing System are the trademark property of Easy Software Products.

DivX is a trademark of Project Mayo.

Citrix, Independent Computing Architecture (ICA), Program Neighborhood, XenApp, XenApp are registered trademarks or trademarks of Citrix Systems, Inc. in the U.S.A. and other countries.

Gemplus is a registered trademark and GemSAFE a trademark of Gemplus.

Linux is a registered trademark of Linus Torvalds in the United States and/or other countries.

Macromedia and Shockwave are registered trademarks of Macromedia, Inc. in the United States and/or in other countries.

Microsoft, MS, Windows, Windows NT, Windows CE, Excel, PowerPoint and Active Directory are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

palmOne and Tungsten are among the trademarks or registered trademarks owned by or exclusively licensed to palmOne, Inc.

PCL is a registered trademark of Hewlett Packard Company in the United States and/or in other countries.

PS/2 and Token Ring are registered trademarks of International Business Machines Corporation.

RealPlayer is a registered trademark of RealNetworks, Inc.

Safesign is a registered trademark of Thales e-Security Ltd.

SAP and SAP product and service names are the trademarks or registered trademarks of SAP AG in Germany and in several other countries.

SICRYPT is a registered trademark of Infineon Technologies Security Solutions GmbH in Germany and/or in other countries.

SSH and SSH product and service names are the trademarks or registered trademarks of SSH Communications Security Ltd.

Sun, Sun Microsystems, Java and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and in other countries.

UNIX is a registered trademark of The Open Group in the U.S.A. and in other countries.

All other trade names we refer to are service marks or trade marks of the respective manufacturers.

This product includes software developed by The X.org Foundation (www.x.org) and its contributors.

This product includes software developed by The XFree86Project, Inc. (www.xfree86.org/) and its contributors.

Privacy policy

Unicon™ and data privacy

Unicon GmbH, Germany ("Unicon") is committed to protecting and respecting your privacy.

This privacy policy sets out the basis on which we will process any personal information that we may collect about you as a visitor to our website or premises or a contact at one of our customers or potential customers, suppliers or potential suppliers or other business partners or in any other cases where we specifically state that this policy will apply.

This privacy policy also sets out how we protect your privacy and your rights in respect of our use of your personal information.

Unicon may have a privacy policy or statement specific to particular local laws, products or services in which case such policy or statement shall supersede or supplement this privacy policy.

What personal information do we collect from you?

We may collect and process the following personal information about you:

- **Personal information that you give us**: This is information about you that you give to us by filling in forms on our website (or other forms that we ask you to complete), giving us a business card (or similar) or corresponding with us by telephone, post, email or otherwise. It may include, for example, your name, address, email address and telephone number; information about your business relationship with Unicon ; and information about your professional role, background and interests.
- **Personal information that our website and other systems collect about you**:
    - If you visit our website it will automatically collect some information about you and your visit, including the Internet protocol (IP) address used to connect your device to the Internet and some other information such as the pages on our site that you visit. This is used to monitor the performance of the website and improve the experience of visitors to the website. Our website may also download "cookies" to your device – this is described in our separate cookie policy. As described further in the cookie policy, you can change the way in which we use cookies by changing your cookie preferences.
    - If you interact with us through social media services, we may be able to access certain information about such interaction, such as whether you have interacted with us through multiple social networks. We make reasonable efforts to ensure that the social media providers have permission from you to allow us to access certain information about you.

Please note that we are not responsible for the manner in which social media service providers handle your personal information that they may collect from you, as this is their responsibility. For instance, our website may use plugins of social media networks such as Twitter, Facebook, Linkedin, and Google+. If you use one of these plugins, these plugins are able to establish a direct connection between your browser and the sites of the respective social media networks. As this transfer takes place directly between your browser and the respective network, Unicon does not have any access, knowledge or control over any data sent or the use of this data within the respective social media networks. If you make use of a social media network and make content available to such networks, this is not covered by this privacy policy. Instead, the terms of use and privacy policies of the respective operators of those social media networks shall apply to any such content. We encourage you to read the privacy policies and terms and conditions of your social media service providers to understand how they handle your personal information.

- Our website may contain links to websites of third parties who are not affiliated to Unicon . If you access such third party websites by clicking on such links, we are not responsible for the way in which such third parties process your personal information that they collect.
- We may make web-based platforms available to our partners and suppliers and such platforms may collect your personal information if you use such platforms. This privacy policy applies to such personal information collected during the registration process and/or the use of such platforms.
- If you exchange emails, telephone conversations or other electronic communications with our employees and other staff members, our information technology systems may record details of those conversations, sometimes including their content.
- Some of our premises have closed circuit TV systems and other security and access management systems which may record you and certain information about your visit if you visit our premises, for security and safety purposes.

- **Other information**: We may also collect some information from other sources. For example:
  - If we have a business relationship with the organisation that you represent, your colleagues or other business contacts may give us information about you such as your contact details or details of your role in the relationship.
  - We sometimes collect information from third party data providers or publicly available sources for anti-money-laundering, export control, credit rating background checking and similar purposes, and to protect our business and comply with our legal and regulatory obligations.

How will we use your personal information?

We may collect, store and use your personal information for the following purposes:

- to operate, manage, develop and promote our business (including our products and services) and, in particular, our relationship with the organisation you represent (if any) and related transactions including, for example:
    - marketing purposes (when we have either gathered prior opt-in consent and/or have a legitimate interest to send you communications which we believe to be relevant and of use to you);
    - warranty processes; product compliance processes;
    - accounting and billing / payment purposes (including to offer financing solutions to customers, together with our finance partners);
    - to operate, administer and improve our website and premises and other aspects of the way in which we conduct our operations;
    - to offer you the services of our online shops based on the terms and conditions of the dedicated web-shop;
    - to provide you with services or information that you may have requested;
    - keep you informed and updated on relevant products or services you may be interested in;
    - to enable you to take part in our online assessments and surveys;
    - to manage and maintain the relationship with investors; and
    - to operate recruiting activities;
- to protect our business from fraud, money-laundering, breach of confidence, theft of proprietary materials and other financial or business crimes;
- to comply with our legal and regulatory obligations and bring and defend legal claims and assert legal rights; and
- if the purpose is directly connected with an assigned purpose previously made known to you.

We may from time to time review your information held in our systems – including the contents of and other information related to your email and other communications with us – for compliance and business-protection purposes as described above. This may include reviews for the purposes of disclosure of information relevant to litigation and/or reviews of records relevant to internal or external regulatory or criminal investigations. To the extent permitted by applicable law these reviews will be conducted in a reasonable and proportionate way and approved at an appropriate level of management. They may ultimately involve disclosure of your information to governmental

agencies and litigation counterparties as described below. Your emails and other communications may also occasionally be accessed by persons other than the member of staff with whom they are exchanged for ordinary business management purposes (for example, where necessary when a staff member is out of the office or has left Unicon).

We will only process your personal information as necessary so that we can pursue the purposes described above and where we have a legal basis for such processing. Where our lawful basis for processing is that such processing is necessary to pursue our legitimate interests, we will only process your personal information where we have concluded that our processing does not prejudice you or your privacy in a way that would override our legitimate interest. In exceptional circumstances we may also be required by law to disclose or otherwise process your personal information.

Disclosure and international transfer of your personal information

We may disclose your personal information, where reasonably necessary for the various purposes set out above:

- to the other members of the Fujitsu group of companies;
- to your colleagues within the organisation that you represent;
- to service providers who host our web services or other information technology systems or otherwise hold or process your information on our behalf, under conditions of confidentiality and security required by law;
- business partners, channel partners, service partner, agents, suppliers and sub-contractors for the performance of any contract we enter into with them or you;
- to a person who takes over our business and assets, or relevant parts of them; or
- in exceptional circumstances:
    - to competent regulatory, prosecuting and other governmental agencies, or litigation counterparties, in any country or territory; or
    - where we are required by law to disclose.

These disclosures may involve transferring your personal information overseas. If you are dealing with us within the European Economic Area, you should be aware that this may include transfers to countries outside the European Economic Area, which have not been determined by the European Commission to have an adequate level of data protection. In those cases, where we transfer your personal information to other members of the Fujitsu group or our service providers, we will ensure that our arrangements with them are governed by relevant legal mechanisms and safeguards

including data transfer agreements, designed to ensure that your personal information is protected, on terms approved for this purpose by the European Commission.

How long do we keep your personal information?

We will delete your personal information when we no longer need such personal information, for instance where:

1. it is no longer necessary for us to retain your personal information to fulfil the purposes for which we had collected it;
2. we believe that your personal information that we hold is inaccurate; or
3. in certain cases where you have informed us that you no longer consent to our processing of your personal information.

Sometimes, however:

1. there are legal or regulatory requirements which may require us to retain your personal information for a specified period, and in such cases we will retain your personal information for such specified period; and;
2. we may need to retain your personal information for certain longer periods for product liability purposes or in relation to legal disputes, and in such cases we will retain it for such longer periods to the extent required.

Note that we may retain some limited information about you even when we know that you have left the organisation that you represent, so that we can maintain a continuous relationship with you if and when we are in contact with you again, representing a different organisation.

What are your rights?

You have the following rights (subject to applicable local laws) in relation to the personal information that we hold about you:

- to access your personal information, and some related information, under relevant data protection law;
- to require any inaccurate personal information to be corrected or deleted;
- to object to our use of your personal information for direct marketing purposes at any time and you may have the right to object to our processing of some or all of your personal information (and require them to be deleted) in some other circumstances;
- to require us to delete your personal information in certain circumstances;

- to require us to restrict or block the processing of your personal information in certain circumstances (when processing is restricted, we can still store your personal information, but may not use it further);
- to obtain from us your personal information, in a structured, commonly used and machine-readable format in certain circumstances. Further, you may have the right to require us to transmit your personal information directly to another person (for instance a new provider) where it is technically feasible to do so; and
- where we are processing your personal information based on your consent to such processing, to withdraw your consent at any time.

Further information and advice about your rights can be obtained from the data protection regulator in your country.

If you wish to exercise any of these rights (subject to applicable local laws), or have complaints about our processing of your personal information, please contact us as set out below. You can also lodge a complaint about our processing of your personal information with the body regulating data protection in your country

Sensitive information

We request you not send to or share with us any sensitive personal information (e.g. information related to racial or ethnic origin, political opinions, religion or other beliefs, health, genetic, or biometric data, criminal background or trade union membership).

Personal information of children

Unicon respects the privacy of children and does not knowingly collect, use or disclose personal information of children under 16 years of age without verifiable consent from a parent or guardian.

If Unicon becomes aware that the information submitted to or collected by Unicon is personal information of children without first receiving verifiable consent from a parent or guardian, Unicon will take prompt steps to delete such personal information.

If a parent or guardian wishes to raise a concern regarding personal information pertaining to a child, please contact us using the details set out in the section titled “Contact Us” below.

Automated decision making

We may use technology which tracks your use of our website and your interactions with us and which helps us to build a profile of your preferred products and information requests. If we do this,

what this means for you is that you are more likely to receive offers and information that are tailored to your specific preferences, based on your previous profile activity.

If we use automated decision making in respect of any specific service or product and such automated decision making produces legal effects concerning data subjects or similarly significantly affects them, we will provide information required by applicable laws in respect of such automated decision making.

We may use automated decision making in the context of comparing delivery and personal data with export control sanction lists of the European Union, the United States of America, Japan and any other country from which an export takes place. Depending on the result of these controls Unicon is obliged to take appropriate measures within the context of export control regulations, which are stipulated by the European Union, the United States of America, Japan and any other country from which the export takes place.

Contact us

If you have any questions, comments and requests regarding this privacy statement or our processing of personal information, please contact us under `<http://www.unicon.com/en/contact/>`.

If you would like to contact our Data Protection Office, please contact us.

Changes to this policy

This privacy policy is valid from the time of its publication on the Unicon websites, and all previous versions of this privacy policy become invalid as of such time. Any changes we make to this privacy statement in the future will be posted to our website and also available if you contact us. Please check back frequently to see any changes.

We will also keep available previous versions of this privacy policy on our website indicating their date and version number.

Last updated: 28 May 2018

Cookie policy

Our website uses “cookies” to distinguish you from other users of our website and to understand how you use our website. This helps us to improve your browsing experience and also to improve our website.

A cookie is a small file of letters and numbers that we store on your browser or the hard drive of your computer if you agree.

You can learn more about our use of cookies, including the types and purpose of the cookies used by our website, by clicking the link for further information in the “cookie notice” which appears as a banner on our website when you first visit our website. Through the cookie notice, you may accept (for the purposes listed), or decline, the use of some or all cookies used by our website. The cookie notice will set out a list of any “essential” cookies which you may not be able to turn off without losing access to some or all parts of our website.

After you have selected your cookie settings through the cookie notice, you will be able to access a “cookie consent” tool on each web page of our website, on each subsequent visit to our website. Clicking on the cookie consent tool will allow you to gain up-to-date information about the cookies being used by our website and manage your consent to some or all of the cookies at any time.

Please consult our Privacy policy to know more about the basis on which we will process any personal information that we may collect about you as a visitor to our website and how we protect your privacy and your rights in respect of our use of your personal information.

If you have any questions, comments and requests regarding this cookie policy, please contact us.

This cookie policy is valid from the time of its publication on the Unicon™ websites, and all previous versions of this privacy policy become invalid as of such time.

Last updated: 28 May 2018

Third party software

Below you will find a collection of links to third-party software and open source projects that are integrated into our software solutions or can be optionally integrated.

For our technology partners, see our home page under unicon.com/partners.

COMPONENT	DESCRIPTION
Avahi	Networking of devices in local networks
CUPS	Print interface with central configuration
FreeRDP	eLuxRDP is based on FreeRDP
Fluendo	Audio/Video codec packages (subject to license)
Gnome	Free Unix/Linux desktop environment
Linux Kernel	Linux kernel
Maliit	Virtual keyboard
MATE-Desktop	Free Unix/Linux desktop environment
Mozilla	Firefox web browser
NoMachine	Remote desktop software
OpenVPN	VPN client
PCSC-Lite	Smart card reader drivers
PulseAudio	Free sound middle-ware
Qt Project	Cross-platform C++ framework
Squid	Free proxy server

COMPONENT	DESCRIPTION
SSH	Secure remote access
VNC	Mirroring/Remote maintenance
WPA Supplicant	Encryption for wireless networks, 802.1X network security
xfree86	X-Windows graphic interface (X11)